Polkuorientoituneen SQL-kyselykielen käyttäjätestaus

Heidi Kari

Tampereen yliopisto
Informaatiotieteiden yksikkö
Tietojenkäsittelyoppi
Pro gradu -tutkielma
Ohjaajat: Marko Junkkari ja
Kati Iltanen
Marraskuu 2014

Tampereen yliopisto
Informaatiotieteiden yksikkö
Tietojenkäsittelyoppi
Heidi Kari: Polkuorientoituneen SQL-kyselykielen käyttäjätestaus
Pro gradu -tutkielma, 81 sivua, 5 liitesivua
Marraskuu 2014

Kyselyt hierarkkisesti organisoidusta relaatiotietokannasta voivat olla SQL-kyselykielellä pitkiä ja monimutkaisia ja vaatia käyttäjältä tietokannan rakenteen tuntemusta. Polkuorientoituneessa PathSQL-kielessä hierarkkiseen dataan kohdistuvia kyselyitä on pyritty helpottamaan korvaamalla liitosehtojen käyttö polkuilmaisuilla. Tässä tutkielmassa tutkitaan käyttäjäkokeen avulla sitä, miten käyttäjät suoriutuvat hierarkkiseen dataan kohdistuvista kyselyistä PathSQL-kielellä verrattuna tavalliseen SQL-kieleen. Lisäksi tutkielmassa analysoidaan koehenkilöiden PathSQL-kyselyissä tekemiä virheitä. Kokeen tuloksen perusteella vaikuttaa siltä, että PathSQL-kieli nopeuttaa kyselyiden tekoa hierarkkisesti organisoidusta datasta, ja että PathSQL-kyselyt tuottavat oikeita vastauksia vähintään yhtä usein kuin SQL-kyselyt.

Avainsanat ja -sanonnat: kyselykieli, SQL, käyttäjäkoe.

Sisällys

1. Johdanto

Kaikkein yleisimmin käytetty tietokantojen kyselykieli on Structured Query Language eli SQL [Chamberlin & Boyce, 1974]. Kieli perustuu Edgar F. Coddin [1970] kehittämään relaatiotietomalliin. Eräs SQL-kielen puutteista on hankala navigointi sellaisissa tietokannoissa, joissa tieto on järjestetty hierarkkisesti. Tässä tutkielmassa tarkastellaan vaihtoehtoista tapaa muodostaa kyselyitä hierarkkiseen dataan relaatiotietokannoissa. Tutkielmassa verrataan perinteistä SQL-kieltä ja polkuorientoitunutta SQL-kieltäsekä analysoidaan käyttäjätestin tuloksia.

Yleisesti hierarkialla tarkoitetaan osittaista järjestystä, jossa asiat sijaitsevat toistensa ylä- tai alapuolella, tai samalla tasolla. Hierarkioita käytetään monenlaisen tiedon esittämiseen ja jäsentämiseen. Esimerkiksi eliöiden, kuten kasvien ja eläinten, tieteellisessä luokittelussa käytetään monitasoista hierarkiaa, joka muodostuu eri kategorioista ja näiden alakategorioista. Monet ihmisen luomat järjestelmät, esimerkiksi erilaiset yhteiskunnalliset organisaatiot kuten valtion hallinto, muodostavat hierarkkisen rakenteen.

Osa-kokonaisuussuhteet voidaan nähdä hierarkiana, jossa suurin kokonaisuus sijaitsee hierarkiassa ylimpänä ja kokonaisuuden muodostavat osat sen alapuolella. Osat voivat muodostua toisista osista, jotka sijaitsevat hierarkiassa alempana. Tällöin monet fyysisen maailman kohteet, kuten eläinten tai kasvien anatomia, tai monista eri osista koostuvat esineet (kuten koneet tai rakennukset), voidaan esittää hierarkioina. Esimerkiksi auto voidaan nähdä kokonaisuutena, joka muodostuu muun muassa pyöristä, rungosta ja moottorista, jotka puolestaan koostuvat muista osista. Osa-kokonaisuussuhteita ovat analysoineet ja luokitelleet tarkemmin esimerkiksi Winston et al. [1987], jotka määrittelevät kuusi erilaista osa-kokonaisuussuhdetyypppiä. Esineen ja sen osan lisäksi osa-kokonaisuussuhde voidaan Winstonin et al. mukaan nähdä kokoelman ja sen jäsenen välillä (esimerkiksi laivasto ja laiva), jonkin kokonaisuuden ja sen osuuden välillä (esimerkiksi piirakka ja pala piirakkaa), aineen ja esineen välillä (kuten metalli ja auto, joka on valmistettu tästä metallista), jonkin toiminnan ja siihen kuuluvan piirteen välillä (kuten ostosten teko ja maksaminen), sekä alueen ja siellä sijaitsevan paikan välillä (kuten Suomi ja Tampere). Motschnig-Pitrik & Kaasbøll [1999] ovat käsitelleet osa-kokonaisuussuhteiden analyysia ja mallintamista olio-orientoituneesta näkökulmasta.

Hierarkkisessa muodossa olevan tiedon säilytykseen ja manipulointiin tietokannoissa on kehitetty erilaisia malleja ja kyselykieliä. Hierarkkisessa tietomallissa data esitetään tietueina (record), jotka ovat vanhempi-lapsi-suhteessa toisiinsa: vanhemman roolissa oleva tietue sijaitsee hierarkiassa lapsitietueen yläpuolella [Elmasri & Navathe, 1989]. Erityisesti osa-kokonaisuussuhteiden muodostamia hierarkioita varten on kehitetty PSE-malli (Part-of Structure Element) [Junkkari, 2005] sekä kyselykieli [Niemi et al., 2004]. Nykyään hyvin suosittu tapa esittää

hierarkkista dataa on XML, johon on kehitetty useita erilaisia kyselykieliä. Oliotietokannoissa hierarkkista dataa voidaan esittää kompleksisten olioiden avulla. Oliotietokannoissa navigointiin on kehitetty erilaisia kyselykieliä ja järjestelmiä, joista tunnetuin on OQL [Cluet, 1998], joka muistuttaa relaatiotietokantojen SQL-kieltä ja on Object Data Management Groupin standardoima [Cattell & Barry, 2000]. Myös erityisesti kompleksisia olioita varten on kehitetty navigointijärjestelmiä [Hua & Tripathy, 1994].

Myös relaatiotietokantoja varten on kehitetty hierarkkisen tiedon organisointiin ja käsittelyyn tarkoitettuja sovelluksia. Esimerkiksi Non-First-Normal-Form eli NF^2 [Roth et al., 1988] on relaatiotietomalli, joka mahdollistaa hierarkkisen datan tallentamisen tietokantaan sisäkkäisinä relaatioina, mutta datan käsittely voi olla tällöin melko monimutkaista. Niemi & Järvelin [1995] ovat kehittäneet käyttöliittymän NF^2-relaatioiden käsittelyn helpottamiseksi.

Kyselyt monimutkaisia hierarkioita muodostavista relaatioista eivät ole perinteisissä SQL-tietokannoissa kovinkaan yksinkertaisia: ne vaativat käyttäjältä usein lukuisten liitosehtojen, operaatioiden ja pahimmassa tapauksessa jopa alikyselyiden käyttöä, sekä relaatiohierarkian rakenteen tarkkaa tuntemusta. Johanna Vainion ja Marko Junkkarin [2014] kehittämällä PathSQL-kyselykielellä pyritään helpottamaan kyselyiden tekoa SQL-tietokantaan tallennetusta hierarkkisesta datasta korvaamalla muun muassa vierasavain-pääavainparien vertailu ja liitosoperaatiot polkuilmaisuilla. Myös käyttäjän tarvetta tuntea tietokantakaavion tarkka rakenne on pyritty vähentämään ja mahdollistamaan tietokannan rakenteen tutkiminen kyselyjen avulla.

Polkuilmaisuja on käytetty aiemmin useissa eri kyselykielissä. Monet XML-kyselykielet, kuten XPath [World Wide Web Consortium, 2014 a], XQuery [World Wide Web Consortium, 2014 b], XIRQL [Fuhr & Großjohann, 2001], XML-QL [Deutsch et al., 1999] ja Lorel [Goldman et al., 1999], ovat polkuorientoituneita. Lisäksi polkuilmaisuja on sovellettu muun muassa oliotietokantojen kyselykielissä [Carey et al., 1988] sekä rakenteettomalle datalle tarkoitetussa UnQL-kyselykielessä [Buneman et al., 1996].

Tässä tutkielmassa on tarkoitus selvittää käyttäjäkokeen avulla, miten käyttäjät suoriutuvat hierarkkiseen dataan kohdistuvista kyselyistä PathSQL-kielellä verrattuna SQL-kieleen. Kysymyksiä, joihin kokeella pyritään löytämään vastauksia, ovat muun muassa se, kummalla kielellä käyttäjät suoriutuvat nopeammin kyselyiden kirjoittamisesta, ja se, kumman kielen käyttäminen tuottaa enemmän oikeita tuloksia.

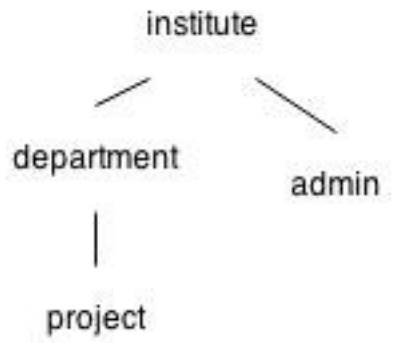
Tutkielman toisessa luvussa esitellään molemmat käyttäjäkokeessa käytetyt kielet, SQL ja PathSQL, sekä esimerkkinä polkuilmaisujen aiemmasta soveltamisesta XML-kyselykielet XPath ja XQuery. Kolmannessa luvussa esitellään aiempia relaatiotietokantojen kyselykieliin kohdistuneita käyttäjätutkimuksia. Neljännessä luvussa kuvataan koeasetelma sekä kerätyn aineiston käsittelyprosessi. Viides luku käsittelee kokeen tuloksia. Kuudennessa luvussa tarkastellaan tarkemmin, millaisia virheitä koehenkilöt tekivät ratkaistessaan tehtäviä PathSQL-kielellä. Tällä

pyritään saamaan tietoa muun muassa siitä, mitkä kielen ominaisuudet ovat mahdollisesti hankalia ja vaikeita käyttäjille. Seitsemännessä luvussa esitetään joitakin ehdotuksia kielen jatkokehitystä tai mahdollisia tulevia käyttäjäkokeita ajatellen.

2. Kyselykielet ja hierarkkinen data

Tässä luvussa keskitytään hierarkioiden käsittelyyn SQL-kielellä ja esitellään PathSQL-kieli, jossa sovelletaan polkuilmausten käyttöä kyselyissä. Polkuilmauksia on sovellettu laajasti hierarkioiden käsitttelyssä XML-kyselykielissä. Tämän vuoksi luvussa käsitellään lyhyesti myös XML-kieltä yleisimpine kyselykielineen.

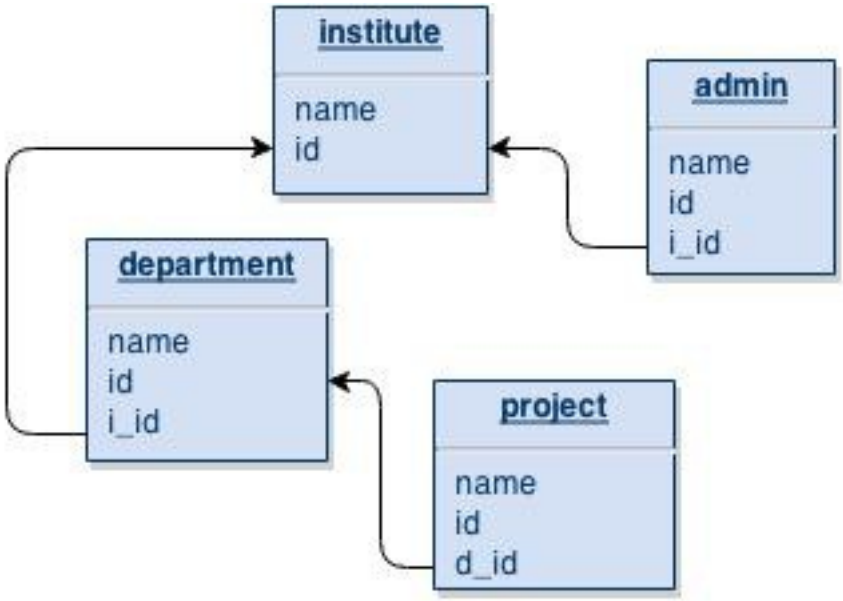
Luvun esimerkeissä käytetään instituution hierarkkista rakennetta (kuva 1). Instituutioilla (institute) voi olla osastoja (department) ja hallintoelimiä (admin). Osastoilla voi puolestaan olla projekteja (project).



Kuva 1. Instituution hierarkkinen rakenne [Vainio & Junkkari, 2014].

2.1. SQL

Relaatiomalli on Edgar F. Coddin [1970] kehittämä tietokantamalli, jossa data esitetään monikkoina (tuple), jotka kuvaavat yksilöitä ja niiden ominaisuuksia. Relaatiot eli taulut ovat joukko keskenään rakenteeltaan samankaltaisia monikoita, joissa monikoiden attribuutit muodostavat taulun sarakkeet. Monikot itsessään muodostavat taulun rivit. Relaatiotietokannoissa suhteet taulujen välillä on määritelty vierasavainten avulla, eli taulussa on omana sarakkeenaan toiseen tauluun tai tauluun itseensä viittaava tunniste. Kyselyt useista tauluista tehdään siis vertaamalla taulujen avaimia toisten taulujen vierasavaimiin. Esimerkiksi kuvan 1 hierarkia voidaan esittää relaatiotietokannassa kuvassa 2 esitetyllä tavalla. Tauluissa department ja admin on attribuutteina tauluun institute viittaavat vierasavaimet (i_id), ja taulussa project on tauluun department viittaava vierasavain (d_id).



Kuva 2. Instituutio aliorganisaatioineen relaatiotietokannassa [Vainio & Junkkari, 2014].

SQL eli Structured Query Language on relaatiotietokantoja varten kehitetty kyselykieli, jolla voi tehdä kyselyjä tietokannoista ja määritellä tietokantojen rakennetta sekä muokata niiden sisältämää tietoa. Kielen esimuoto, SEQUEL eli Structured English Query Language, kehitettiin 1970-luvun alussa [Chamberlin & Boyce, 1974], ja SQL-kieli standardoitiin 1980-luvulla.

SQL-kyselyt sisältävät aina SELECT- ja FROM-lauseet. SELECT-lauseessa ilmaistaan, mitä taulujen sarakkeita kyselyn tulokseen halutaan. Lauseessa voi olla myös tuloksen muokkaukseen liittyviä funktioita, kuten aggregointifunktioita. FROM-lauseessa ilmaistaan, mihin tauluihin haku kohdistuu. Näiden lisäksi kyselyssä voi olla muitakin lauseita, joilla muokataan haun tuottamaa tulostaulua. Esimerkiksi WHERE-lauseessa määritellään, millä ehdoilla taulujen rivit sisällytetään tulokseen. WHERE-lause voi sisältää erilaisia operaattoreita (kuten vertailuoperaattorit ja loogiset operaattorit), joilla tulosrivien joukkoa rajoitetaan. Kysely voi koostua myös useista kyselyistä. Usean kyselyn tuloksia voi yhdistää esimerkiksi joukko-operaattoreilla tai alikyselyjen yhteydessä liitoksilla. Kyselyt voivat olla sisäkkäisiä, jolloin jossakin pääkyselyn osassa käytetään alikyselyn tuottamaa tulosta.

Kuvan 2 tietokanta vastaa kuvan 1 hierarkiaa. Esimerkiksi kaikki vähintään yhden projektin sisältävien instituutioiden nimet ja projektien nimet saadaan kuvan 2 tietokannasta esimerkin 1 kyselyllä.

Esimerkki 1

```
SELECT institute.name, project.name
FROM institute, department, project
WHERE insitute.id=department.i_id AND department.id=project.d_id
```

Kyselyn voi tehdä myös käyttämällä liitosoperaatioita, joissa useita tauluja voidaan liittää FROM-osassa yhdeksi tauluksi. Michael M. David [2003] on ehdottanut LEFT OUTER JOIN -liitosten käyttöä hierarkkisissa kyselyissä. Esimerkin 1 kysely voidaan korvata liitoksia käyttäen esimerkin 2 kyselyllä, jossa liitetään yhteen taulut institute, department ja project sekä valitaan näistä tulokseen instituutioiden ja niihin mahdollisesti kuuluvien projektien nimet. Tällöin kyselyn tulokseen saadaan mukaan myös sellaiset instituutiot, joilla ei ole ollenkaan projekteja.

Esimerkki 2

```
SELECT institute.name, project.name
FROM institute
LEFT OUTER JOIN department ON institute.id=department.i_id
LEFT OUTER JOIN project ON department.id=project.d_id
```

Kyselyt monista tauluista ja monimutkaisista tauluhierarkioista voivat olla vaativia varsinkin kokemattomille käyttäjille, sillä ne vaativat yleensä useiden liitosehtojen, joukko-operaatioiden tai alikyselyjen käyttöä. Lisäksi pitkien kyselyiden kirjoittaminen vie aikaa kokeneemmiltakin käyttäjiltä, ja pitkissä kyselyissä mahdollisuus tehdä kirjoitusvirheitä kasvaa.

2.2. XML-kyselykielet

Extensible Markup Language eli XML on tiedon kuvaamiseen kehitetty merkintäkieli, joka on World Wide Web Consortiumin standardoima [World Wide Web Consortium, 2008]. XML on järjestelmästä riippumatonta ja sekä ihmisten että koneiden luettavissa, joten se on suosittu tapa varastoida ja siirtää dataa. XML perustuu XML-tietomalliin [World Wide Web Consortium, 2005].

XML-dokumentti koostuu elementeistä, joita merkitään alku- ja loppumerkillä (tag). Näiden merkkien väliin jää elementin sisältö. Esimerkiksi <institute>uta</institute> kuvaa institute-elementin ja sen sisällön (uta). Elementit voivat sisältää toisia elementtejä. Elementit muodostavat puurakenteen, jolla on aina juuri. Juuri on elementti, joka sisältää muut mahdolliset elementit.

Hierarkkisten suhteiden esittäminen XML-kielellä on luontevaa, sillä XML-dokumentit muodostavat puurakenteen, jossa elementit voivat sisältää alielementtejä. Relaatiotietokannan kaltaista vierasavainten käyttöä ei siis elementtien välillä tarvita. Esimerkiksi kuvan 2 tietokannan

sisältämän yksittäisen instituution tiedot voitaisiin esittää esimerkin 3 XML-dokumentin mukaisesti.

Esimerkki 3

```
<institute>
    <name>uta</name>
    <admin>
        <name>aktuaari</name>
    </admin>
    <admin>
        <name>kirjaamo</name>
    </admin>
    <department>
        <name>sis</name>
        <project>
            <name>ngis</name>
        </project>
        <project>
            <name>trix</name>
        </project>
    </department>
    <department>
         ...
    </department>
</institute>
```

XML-datan käsittelyä varten on kehitetty useita kieliä. Tunnetuin näistä on XPath, joka on World Wide Web Consortiumin kehittämä polkuorientoitunut kyselykieli [World Wide Web Consortium, 2014 a]. XPath-kielessä XML-dokumentin puurakennetta kuvataan polkuina. Polussa siirtymät puun eri solmujen välillä kuvataan yksinkertaisimmillaan pelkällä vinoviivalla. Tämä on lyhennetty esitystapa XPath-kielen varsinaisesta syntaksista, jossa määritellään sanallisesti jokaisen siirtymän akseli (esimerkiksi child tarkoittaa lapsielementtiä, descendant mitä tahansa jälkeläiselementtiä ja parent vanhempaa). Esimerkiksi polkuilmaisu /child::A/child::B/child::C voidaan lyhentää muotoon /A/B/C. Polku /institute/department/project tuottaa tulokseksi esimerkin 3 juurielementin institute lapsielementin department lapsielementin project. Sama tulos saadaan myös kyselyllä /institute//project, joka tuottaa tulokseksi kaikki elementin institute

jälkeläiselementit project. Käyttäjän ei tarvitse määritellä erikseen hierarkiassa näiden kahden elementin välissä mahdollisesti olevia elementtejä, ja elementit voivat sijaita missä tahansa kohdassa hierarkiaa elementin institute alapuolella. XPath sisältää erilaisia operaattoreita, jotka mahdollistavat navigoinnin puuhierarkiassa sekä aritmeettiset operaatiot, vertailuoperaatiot ja loogiset operaatiot. Esimerkiksi asteriskilla voidaan viitata mihin tahansa elementtiin: polkuilmaisu /institute//* tuottaa tulokseksi kaikki elementin institute jälkeläiset. Pisteellä voidaan taas viitata elementtiin itseensä: esimerkiksi ilmaisulla institute[./department and ./admin] saadaan elementin institute lapsielementit department ja admin. Myös piste ja merkintä // ovat lyhennettyjä muotoja XPath-kielen varsinaisesta syntaksista.

Toinen tunnettu XML-kyselykieli on World Wide Web Consortiumin kehittämä XQuery [World Wide Web Consortium, 2014 b], jolla voidaan tehdä kyselyitä XML-datasta. Koska XQuery on ilmaisuvoimaltaan ohjelmointikieli, se soveltuu monipuoliseen XML-datan käsittelyyn. Kyselyt voivat sisältää tyypillisen ohjelmointikielen tavoin muuttujia, ehtolausekkeita ja silmukoita. XQuery-kyselyt voivat sisältää myös XPath-ilmaisuja, sillä XQuery on yhteensopiva XPath-kielen kanssa. Molemmat kielet perustuvat XML- tietomalliin.

XQuery-kyselyt muistuttavat rakenteeltaan SQL-kyselyitä. Ne jaetaan viiteen eri lohkoon, jotka noudattavat aina samaa järjestystä: FOR, LET, WHERE, ORDER BY ja RETURN. Kyselyssä on aina vähintään yksi FOR- tai LET-lohko. Niitä voi olla useita peräkkäin. FOR-lohkossa käydään silmukassa läpi yksi kerrallaan useita muuttujia. Esimerkiksi jonkin dokumentin kaikki elementit voidaan käydä läpi FOR-silmukassa. LET-lohkossa esitellään muuttujia, joita ei käydä läpi silmukassa vaan käsitellään yhtenä kokonaisuutena. Esimerkiksi jonkin dokumentin kaikki elementit voidaan liittää yhdeksi muuttujaksi. WHERE-lohkossa voidaan määritellä ehtoja kyselyn tulokselle, ja ORDER BY -lohkossa kyselyn tulos voidaan järjestää. WHERE- ja ORDER BY -lohkot voivat puuttua kyselystä, eli ne eivät ole pakollisia. RETURN-lohkossa määritellään se, mitä kyselyn tulee palauttaa. RETURN-lohko on pakollinen osa kyselyä. Esimerkiksi kysely FOR $x in //admin RETURN $x palauttaa yksitellen kaikki esimerkin dokumentin admin-elementit sisältöineen, kun taas kysely LET $x in //admin RETURN $x palauttaa kaikki esimerkin 3 dokumentin admin-elementit sisältöineen yhtenä sekvenssinä.

2.3. PathSQL

Johanna Vainion ja Marko Junkkarin [2014] kehittämä PathSQL on SQL-kieleen perustuva kyselykieli, jossa integroidaan polkuilmaisut ja relaatiotietomalli. PathSQL on syntaksiltaan SQL:n kaltainen kyselykieli, jossa liitosoperaatiot on korvattu polkuilmaisuilla. Taulujen välisten suhteiden käsittely on siis osittain samankaltaista kuin XPath- ja XQuery-kielissä: vierasavain-pääavainparien vertailua ei tarvita vaan taulujen muodostamassa hierarkiassa jokainen siirtymä merkitään vinoviivalla. PathSQL-kyselyissä polku liitetään aina kyselyn FROM-osaan. Esimerkiksi kaikki

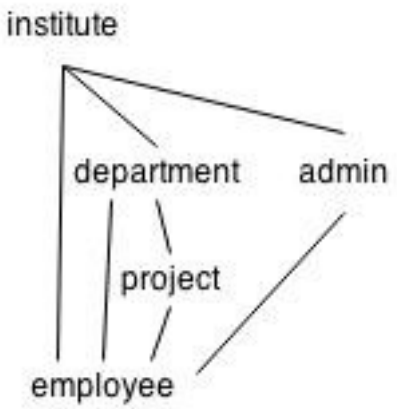
instituutioiden ja niihin mahdollisesti liittyvien projektien nimet saadaan kuvan 2 tietokannasta esimerkin 4 kyselyllä.

Esimerkki 4

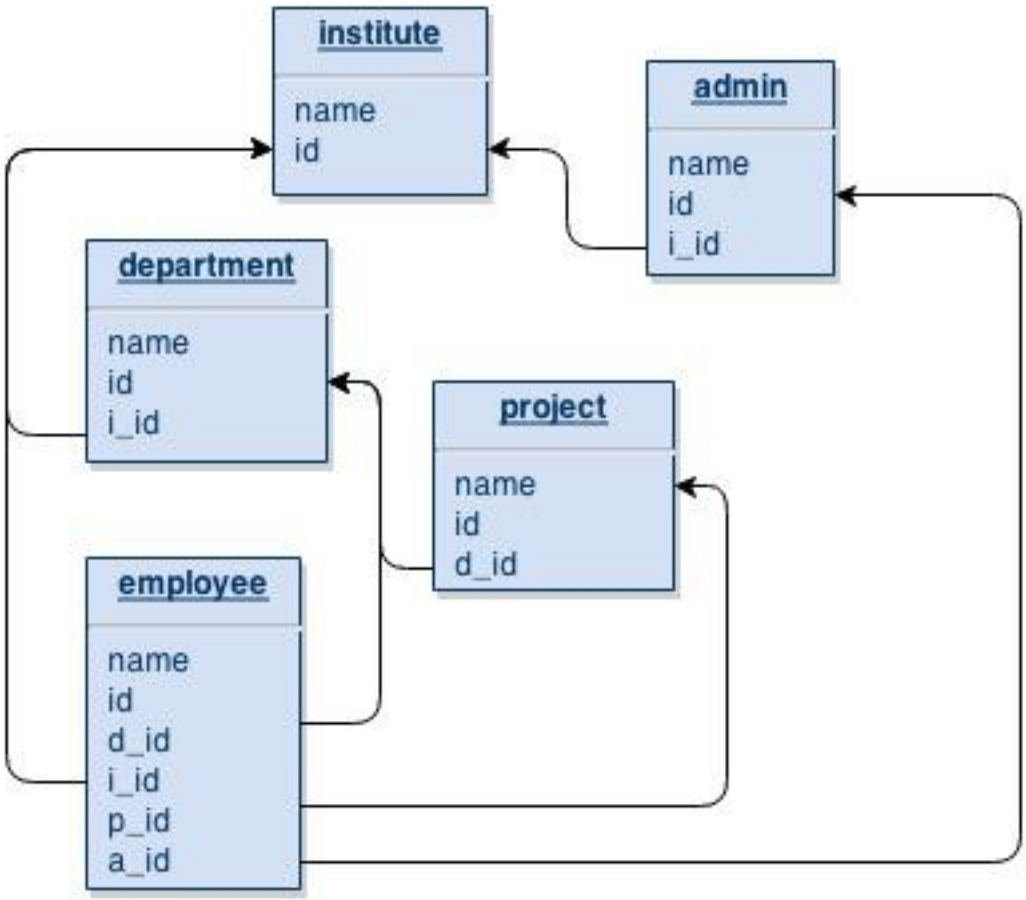
```
SELECT institute.name, project.name
FROM institute/department/project
```

Sama tulos saadaan myös käyttämällä kyselyn FROM-osassa polkua institute//project, joka tuottaa tulokseksi kaikki instituutioiden (institute) alapuolella kuvan 1 hierarkiassa olevien projektien (project) nimet riippumatta siitä, mitä tauluja taulujen institute ja project välissä mahdollisesti on. Käyttäjän ei ole siis tässä tapauksessa edes välttämätöntä tuntea tarkalleen tietokannan hierarkkista rakennetta.

Asteriskilla voidaan viitata mihin tahansa tauluun hierarkiassa samaan tapaan kuin XPath-kielessä: esimerkiksi polku institute/*/project tuottaa tulokseksi kaikki sellaiset projektit, joita ylempänä hierarkiassa on yksi taulu ennen instituutiota.



Kuva 3. Instituution laajennettu hierarkkinen rakenne [Vainio & Junkkari, 2014].



Kuva 4. Instituution rakenne tietokantakaaviona, johon lisätty relaatio työntekijöille [Vainio & Junkkari, 2014].

Kuvassa 3 esitettyyn instituutioiden rakennetta kuvaavaan hierarkiaan on lisätty työntekijä (employee). Työntekijä voi työskennellä suoraan instituutiossa, siihen kuuluvassa hallintoelimessä (admin), osastolla (department) tai jossain osaston projektissa (project). Tätä hierarkiaa vastaa kuvan 4 tietokanta. Useaan hierarkian haaraan kohdistuvat kyselyt muistuttavat rakenteeltaan XPath-kielen ilmaisuja: esimerkiksi polku institute(/department, /admin) tuottaa tulokseksi välittömästi jonkin instituution alapuolella hierarkiassa olevat osastot (department) ja hallintoelimet (admin). Näitä kielen eri ominaisuuksia voi yhdistää samaan kyselyyn. Esimerkin 5 kysely tuottaa tulokseksi kaikkien projektien nimet, jotka sijaitsevat instituutioiden alapuolella hierarkiassa, sekä kaikkien instituutioon kuuluvien ylläpitäjien nimet.

Esimerkki 5

```
SELECT institute.name, project.name, admin.name
FROM institute(//project, /admin)
```

Kyselyn tulos:

institute name	project name	admin name
uta	ngis	aktuaari
uta	ngis	kirjaamo
uta	trix	aktuaari
uta	trix	kirjaamo

PathSQL:n syntaksi voidaan esittää laajennetulla BNF-notaatiolla (EBNF) [ISO/IEC 14977, 1996] alla olevalla tavalla. Polku alkaa aina relaation nimellä (r_name), jota seuraa polun loppuosa (tail). Loppuosa voi olla lineaarinen tai sarjallistettu. Lineaarisen loppuosan alussa on erotin (separator), joka voi olla joko vinoviiva tai kaksi peräkkäistä vinoviivaa. Erottimen jälkeen tulee relaation nimi-ilmaisu (r_name_expr), joka voi olla joko relaation nimi tai asteriski. Tätä voi seurata toinen loppuosa (tail). Sarjallistettu loppuosa ilmaistaan haarailmaisuna (b_expr), jonka alussa ja lopussa on aina sulkumerkki. Välissä on aina vähintään kaksi loppuosaa (tail), mutta niitä voi olla enemmänkin. Loppuosat on erotettava toisistaan pilkulla. Alla oleva esitys vastaa alkuperäistä Vainion ja Junkkarin [2014] EBNF-esitystä lukuun ottamatta haarailmaisussa b_expr vaadittua polun loppuosien määrää. Alkuperäisessä esityksessä loppuosia vaadittiin vähintään yksi, kun taas tässä työssä käytetty PathSQL-järjestelmä vaatii b_expr-ilmaisuun vähintään kaksi loppuosaa. Yksi polku sulkumerkkien sisällä aiheuttaa siis syntaksivirheen.

```
path_expr ::= r_name tail
tail :: = (separator r_name_expr [tail]) | b_expr
b_expr ::= "(" tail, tail {, tail} ")"
r_name_expr ::= r_name | "*"
separator ::= "/" | "//"
```

PathSQL:n toteutus perustuu SQL-kieleen, eli polkuilmaisut muutetaan kyselyä suoritettaessa SQL-kielen ilmaisuiksi. Relaatiot yhdistetään tulostauluksi Davidin [2003] lähestymistapaan perustuen LEFT OUTER JOIN -liitosoperaatiolla. Tällöin yhden vinoviivan käyttö merkitsee yhtä

liitosoperaatiota: esimerkiksi polkuilmaisu institute/department muutetaan SQL-ilmaisuksi institute LEFT OUTER JOIN department ON institute.id=department.i_id. Kaksi vinoviivaa relaatioiden välissä sisältää kaikki mahdolliset polkuilmaisut alkurelaatiosta loppurelaatioon ja nämä puolestaan muutetaan SQL:n LEFT OUTER JOIN -liitosoperaatioiksi. Tässä työssä käytetyssä PathSQL-järjestelmässä näiden kaikkien tuottamat tulokset yhdistetään yhdeksi tulostauluksi UNION-operaation avulla. Kuvan 4 tietokantaan kohdistuvan kyselyn polkuilmaisu institute//employee sisältää esimerkin 6 polut.

Esimerkki 6

- `institute/employee`
- `institute/admin/employee`
- `institute/department/employee`
- `institute/department/project/employee`

Sulkumerkinnän sisällä luetellut polun haarat liitetään LEFT OUTER JOIN -operaatioilla. Esimerkiksi ilmaisu institute(/department,/admin) tuottaa esimerkin 7 liitosoperaatiot sisältävän SQL-kyselyn.

Esimerkki 7

```
institute LEFT OUTER JOIN department ON
institute.id=department.i_id
LEFT OUTER JOIN admin ON institute.id=admin.i_id
```

PathSQL-järjestelmässä kyselyn WHERE-osassa voi viitata vain FROM-osan polussa annettujen taulujen sarakkeisiin. Jos kyselyyn, jossa käytetään useamman kuin yhden polun tuottavaa //-ilmaisua, lisätään WHERE-osassa ehtoja, ehdot liitetään kaikkiin tuotettuihin polkuihin. Tämä voi aiheuttaa SQL-virheitä sellaisissa tapauksissa, joissa vertailussa käytettyä taulua ei löydy kaikista kyselyn poluista.

Esimerkki 8

```
SELECT employee.name FROM institute//employee WHERE
admin.name='kirjaamo'
```

Esimerkin 8 kyselyn FROM-osa tuottaa esimerkissä 6 annetut polut, ja kun WHERE-osan ehdot liitetään näihin kaikkiin polkuihin, syntyy useita SQL-virheitä. Esimerkiksi polusta

institute/employee ei löydy taulua admin, joten ehdon admin.name='kirjaamo' liittäminen polkuun aiheuttaa virheen.

3. Kyselykielten kokeellinen tutkimus

Relaatiotietokantojen kyselykieliä on tutkittu käyttäjäkokeiden avulla 1970-luvulta alkaen. Monissa tutkimuksissa on tarkasteltu eroja eri kielten välillä, sekä sitä, miten koehenkilöiden ominaisuudet, kuten aiempi tietotekninen kokemus, vaikuttavat tulokseen. Myös muunlaisia kyselykieliä on tutkittu kokeellisesti: esimerkiksi Graaumans [2005] ja Weiand [2010] ovat molemmat tutkineet väitöskirjoissaan eroa kahden XML-kyselykielen välillä koehenkilöitä käyttäen, ja Sengupta ja Dillon [2006] ovat suorittaneet käytettävyyskokeen kehittämälleen Query by Templates -nimiselle XML-kyselykielelle. Tässä luvussa keskitytään kuitenkin SQL-kieleen sekä muihin relaatiotietokantojen kyselykieliin liittyvään tutkimukseen, sillä se liittyy olennaisimmin tämän tutkielman aiheeseen.

3.1. Tutkimusten luokittelu

Reisner [1981] käy läpi kyselykielten kokeellista tutkimusta 1970-luvulta alkaen. Näissä tutkimuksissa on pyritty tutkimaan kyselykielten helppokäyttöisyyttä (ease-of-use). Helppokäyttöisyyden määrittely ja mittaaminen ei Reisnerin mukaan ole yksinkertaista: kyselykielet ovat monimutkaisia ja vaativat monia kognitiivisia toimintoja kuten oppimista, ymmärtämistä ja muistamista.

Tutkijat ovat pyrkineet mittaamaan helppokäyttöisyyttä erilaisten tehtävien avulla. Reisner listaa kuusi erilaista tehtävätyyppiä (taulukko 1). Eniten tutkimuksissa käytetty tehtävätyyppi on Reisnerin mukaan kyselyn kirjoittaminen.

Taulukko 1. Reisnerin [1981] kuvaamat tehtävätyypit.

Tehtävätyyppi	**Kuvaus**
Kyselyn kirjoittaminen	Käyttäjien tulee muodostaa kyselykielinen kysely luonnollisella kielellä olevan kysymyksen pohjalta.
Kyselyn lukeminen	Käyttäjien täytyy kääntää kyselykielinen kysely luonnolliselle kielelle.
Kyselyn tulkinta	Käyttäjät saavat tulostetun tietokannan sisällön, josta heidän tulee etsiä vastaus kyselykieliseen kyselyyn.
Kyselyn ymmärtäminen	Käyttäjät saavat tulostetun tietokannan sisällön, josta heidän tulee etsiä vastaus luonnollisella kielellä olevaan kysymykseen.
Muistaminen	Käyttäjien tulee opetella ja muistaa kokeessa jokin tietokanta.
Ongelmanratkaisu	Käyttäjille annetaan tietokanta ja jokin ongelma. Käyttäjien tulee muodostaa luonnollisella kielellä olevia kysymyksiä, jotka ratkaisevat ongelman. Kysymyksiin tulee löytyä vastaus tietokannasta.

Tehtäviä voidaan käyttää Reisnerin mukaan erityyppisissä kokeissa (taulukko 2). Eri koetyypeillä voidaan Reisnerin mukaan tutkia helppokäyttöisyyden eri puolia. Yleisimmäksi koetyypiksi Reisner mainitsee loppukokeen.

Taulukko 2. Reisnerin [1981] kuvaamat koetyypit.

Koetyyppi	Kuvaus
Oppimisen loppukoe	Järjestetään opetuksen jälkeen. Kokeen tarkoitus on tutkia kielen oppimisen helppoutta.
Välitön ymmärrys	Järjestetään, kun käyttäjälle on opetettu jokin kielen funktio, jota käyttäjän tulee käyttää kokeessa. Kokeen tarkoitus on selvittää tiettyjä kielen oppimisen ongelmia
Kokeet opetuksen aikana	Järjestetään opetuksen aikana. Käyttäjän tulee tietää, mitä tähän asti opituista funktioista tulee käyttää kokeessa. Kokeen tarkoitus on selvittää tiettyjä kielen oppimisen ongelmia.
Tuottavuus	Kokeilla tutkitaan “taitavien” käyttäjien kyselykielen käyttöä. Kokeen tarkoitus on selvittää, kuinka hyvin käyttäjät käyttävät kieltä sen jälkeen, kun ovat saavuttaneet jonkin tietyn oppimisen tason.
Muistaminen	Kokeella tutkitaan sitä, kuinka hyvin koehenkilöt osaavat käyttää opittua kyselykieltä kun he ovat olleet käyttämättä sitä jonkin aikaa. Kokeen tarkoitus on siis testata kyselykielen muistamisen helppoutta.
Uudelleenoppiminen	Kokeella tutkitaan sitä, kuinka helppoa kyselykielen uudelleenoppiminen on koehenkilöille, jotka ovat olleet käyttämättä kieltä jonkin aikaa ja unohtaneet osan siitä.

Reisner erottelee kaksi erilaista kyselykielten tutkimustyyppiä: yksittäisen kyselykielen evaluointi ja kahden tai useamman kyselykielen vertailu. Lisäksi kokeellisilla tutkimuksilla voidaan tukea kyselykielten suunnittelua. Kokeiden tulosten perusteella voidaan tunnistaan kyselykielten ongelmat, selvittää niiden syyt ja esittää niihin parannusehdotuksia. Ongelmien tunnistamisen keinoina Reisner mainitsee esimerkiksi käyttäjien tekemien pienten virheiden (esimerkiksi kirjoitusvirheet) luokittelun sekä käyttäjille hankalien funktioiden tunnistamisen. Ongelmien syiden selvittämiseen voidaan Reisnerin mukaan tehdä esimerkiksi virheanalyysejä, käyttäjäkyselyitä tai funktioiden käytön onnistumisen vertailua eri tyyppisissä kokeissa.

3.2. Kyselykielten kokeellisia tutkimuksia

SEQUEL ja SQUARE

Reisner ja muut [1975] vertailivat tutkimuksessaan SEQUEL- ja SQUARE-kyselykielten käytön oppimista käyttäen kahta koehenkilöryhmää: opiskelijoita, joilla ei ollut ollenkaan aiempaa ohjelmointikokemusta, ja opiskelijoita, jotka olivat suorittaneet vähintään yhden ohjelmointikurssin. Tutkimuksen tarkoituksena oli selvittää, soveltuvatko kielet myös ohjelmointitaidottomien ammattikäyttäjien käytettäväksi.

Koehenkilöitä oli yhteensä 64 ja heidät jaettiin neljään ryhmään: kaksi ohjelmoijioiden (programmer) ryhmää, joista toiselle opetettiin SEQUEL-kieltä ja toiselle SQUARE-kieltä, sekä kaksi ohjelmointitaidottomien (non-programmer) ryhmää, joista toiselle opetettiin SEQUEL-kieltä ja toiselle SQUARE-kieltä. Opetus oli perinteistä luokkaopetusta. Ohjelmoijia opetettiin yhteensä 12 tuntia ja ohjelmointitaidottomia 14 tuntia. Koehenkilöiden osaamista testattiin pienemmillä kokeilla kielten opetustunneilla ja laajemmalla loppukokeella, joissa kaikissa koehenkilöt saivat käyttää kurssimateriaaleja ja muistiinpanoja apunaan. Viikon kuluttua loppukokeesta järjestettiin muistikoe, jossa ei saanut olla mukana muistiinpanoja tai muuta kurssimateriaalia. Kokeet olivat kaikille koehenkilöryhmille samanlaiset: niissä oli englanninkielisiä kysymyksiä, jotka tuli muotoilla joko SEQUEL- tai SQUARE-kyselyksi, joka tuottaa tulokseksi vastauksen kysymykseen. Loppukoe ja muistikoe sisälsivät kumpikin 40 kysymystä, joissa testattiin sekä kielten peruspiirteiden hallintaa erikseen että taitoa yhdistellä niitä yhdessä kyselyssä.

Koehenkilöiden vastaukset luokiteltiin viiteen luokkaan niissä esiintyvien virheiden vakavuusjärjestyksessä: täysin oikea vastaus, pieni datavirhe, pieni kielivirhe, sisältövirhe (muodoltaan kieliopin mukainen kysely, joka tuottaa väärän tuloksen) ja muotovirhe. Vastaukset pisteytettiin luokkiensa mukaan, ja jos vastauksesssa esiintyi useita eri virhetyyppejä, ne luokiteltiin vakavimman virheen perusteella.

Kokeiden tulosten perusteella ohjelmoijat oppivat molemmat kielet nopeammin ja paremmin kuin ohjelmointitaidottomat, mutta ohjelmointitaidottomista SEQUEL-kieltä opiskellut ryhmä menestyi paremmin kuin SQUARE-kieltä opiskellut ryhmä.

Query By Example

Thomas ja Gould [1975] tutkivat ohjelmointitaidottomien henkilöiden Query By Example -kyselykielen oppimista. Koehenkilöitä oli yhteensä 39 ja he olivat joko lukio- tai yliopisto-opiskelijoita, joista ainoastaan neljällä oli aiempaa vähäistä ohjelmointikokemusta. Koeryhmiä oli neljä, ja jokaiselle ryhmälle järjestettiin kaksi Query By Example -kielen opetustilaisuutta. Molempien tilaisuuksien päätteeksi järjestettiin koe, joista ensimmäisessä testattiin kielen

helpompien perusominaisuuksien osaamista ja toisessa vaikeampien ominaisuuksien yhdistämistä kielen perusominaisuuksien käyttöön. Kahden viikon päästä kuusi koehenkilöä osallistui uusintakokeeseen, jota seurasi tunnin mittainen kertausluento ja toinen uusintakoe. Kaikissa kokeissa kysymyksiä oli 20 ja jokaisessa niistä koehenkilön tuli muotoilla englanninkielisen kysymyksen pohjalta Query By Example -kysely, jonka tulos tuottaa vastauksen alkuperäiseen kysymykseen. Koehenkilöt merkitsivät vastauksiinsa myös tehtävään käytetyn ajan ja sen, kuinka varmoja he olivat (asteikolla 1-5) vastauksensa oikeellisuudesta.

Koehenkilöiden tehtävien vastauksista keskimäärin 67% oli oikein. Varmuus vastauksen oikeellisuudesta korreloi sen todellisen oikeellisuuden kanssa, mutta suoritusajan ja vastauksen oikeellisuuden välillä ei ollut yhteyttä.

Query By Example -kieltä ovat tutkineet käyttäjäkokeella myöhemmin myös ainakin Yen ja Scamell [1993], jotka vertailivat Query By Examplen ja SQL:n eroa kahdessa erilaisessa koetilanteessa: kynällä ja paperilla tehtävässä kokeessa sekä varsinaisissa tietokoneella tehtävissä tietokantakyselyissä. Koehenkilöt olivat yliopisto-opiskelijoita, joilla oli aiempaa kokemusta tietokoneiden käytöstä mutta ei kyselykielistä. Koehenkilöitä muodostettiin kaksi ryhmää, joista toisessa oli 30 ja toisessa 35 henkilöä. Koehenkilöistä kerättiin taustatietoa kuten aiempi menestys kursseilla ja tietotekninen kokemus.

Kokeen ensimmäisessä vaiheessa molemmille ryhmille opetettiin kokeessa käytettävän tietojärjestelmän käyttöä. Lisäksi ensimmäiselle ryhmälle opetettiin SQL-kyselykielen ja toiselle ryhmälle Query By Example -kielen käyttöä. Molemmat ryhmät tekivät oppimallaan kyselykielellä kokeita sekä kynällä ja paperilla että tietojärjestelmällä. Kokeiden jälkeen he vastasivat tyytyväisyyskyselyyn käyttämästään kielestä. Ensimmäinen vaihe kesti neljä viikkoa, ja viikon tauon jälkeen alkoi kokeen toinen vaihe, jossa ensimmäiselle ryhmälle, jolle oli aiemmin opetettu SQL:n käyttöä, opetettiin Query By Examplen käyttöä, ja toiselle ryhmälle, jolle oli aiemmin opetettu Query By Examplen käyttöä, opetettiin SQL:n käyttöä. Jälleen molemmat ryhmät tekivät myös kokeita kynällä ja paperilla sekä tietojärjestelmällä ja vastasivat kokeiden jälkeen tyytyväisyyskyselyihin. Molemmissa vaiheissa ryhmiä opetettiin erikseen ja opetus oli tavallista luokkaopetusta. Opetukseen sisältyi kaksi yhteensä kolmen tunnin pituista luentoa sekä kotitehtäviä. Opiskelijat saivat myös kielen perusteita käsittelevän ohjekirjan. Kokeen kolmannessa vaiheessa koehenkilöt saivat valita, kummalla kyselykielellä suorittavat kokeet, ja heiltä pyydettiin kyselylomakkeella perustelu valinnalleen. Kokeissa oli sekä yksinkertaisia että monimutkaisia kyselyitä. Kokeiden vastausten virheellisyys arvioitiin samoin kuin aiemmin mainitussa Reisnerin et al. [1975] tutkimuksessa ja vastaukset pisteytettiin niiden oikeellisuuden perusteella.

Tutkimuksessa havaittiin muun muassa se, että kynällä ja paperilla tehtävissä kokeissa koehenkilöt onnistuivat tehtävissä paremmin Query By Example -kielellä, mutta tietojärjestelmää eli oikeaa tietokantaa käytettäessä eroja kielten välillä ei ollut.

SQL ja TABLET

Welty ja Stemple [1981] tutkivat SQL ja TABLET-kyselykielten opittavuutta. Suurin eroavaisuus kielten välillä on Weltyn ja Stemplen mukaan se, että TABLET on proseduraalisempi kuin SQL.

Kokeita järjestettiin kaksi. Ensimmäiseen kokeeseen osallistui 72 yliopisto-opiskelijaa, joista noin puolet oli suorittanut jonkin ohjelmointikurssin. Koehenkilöt jaettiin kahteen ryhmään siten että molemmissa ryhmissä oli tasaisesti sekä ohjelmointikurssin suorittaneita että ohjelmointitaidottomia henkilöitä. Toiseen kokeeseen osallistui 78 opiskelijaa, joista kukaan ei ollut suorittanut yhtään ohjelmointikurssia. Myös nämä koehenkilöt jaettiin kahteen ryhmään. Molemmissa kokeissa toiselle ryhmälle opetettiin SQL-kieltä ja toiselle TABLET-kieltä. Opetuksessa käytettiin oppaita, joita opiskelijat käyttivät itseopiskeluun, sekä 14 korkeintaan 50 minuutin luokkatapaamista, joissa ohjaaja vastasi opiskelijoiden kysymyksiin ja järjesti välikokeita opetetuista asioista. Kurssin jälkeen järjestettiin loppukoe, jossa sai olla mukana kurssimateriaaleja. Kokeessa oli 30 englanninkielistä kysymystä, joiden pohjalta piti muotoilla kysely, joka tuotti vastauksen kysymykseen. Kymmenen näistä oli helppoja, kielen perusasioita käsitteleviä kysymyksiä ja loput 20 vaikeampia, kurssin loppupuolella opetettuja edistyneempiä kielen toimintoja käsitteleviä kysymyksiä. Kolme viikkoa loppukokeen jälkeen järjestettiin muistikoe, jossa ei saanut käyttää kurssimateriaalia muistin tukena. Lisäksi opiskelijoita pyydettiin olemaan opiskelematta koetta varten. Muilta osin muistikoe oli samankaltainen kuin loppukoe.

Kokeiden vastauskyselyt luokiteltiin yhdeksään luokkaan virheellisyyden perusteella virheen vakavuusjärjestyksessä: täysin oikein, pieni kielivirhe, pieni operandivirhe, pieni sisältövirhe, kielen kääntäjän korjattavissa oleva virhe, suuri sisältövirhe (kysely on kielen syntaksin mukainen mutta tuottaa väärän vastauksen), suuri kielivirhe (syntaksivirhe), keskeneräinen kysely, ja puuttuva ratkaisuyritys. Neljä ensimmäistä luokkaa luokiteltiin oikeiksi vastauksiksi ja viisi viimeistä vääriksi vastauksiksi. Jos kyselyssä oli useampi virhe, se luokiteltiin vakavimman virheensä mukaan.

Weltyyn ja Stemplen mukaan oikeiden vastausten osuus oli suurempi vaikeissa koetehtävissä TABLET-kieltä opiskelleilla koehenkilöillä kuin SQL-kieltä opiskelleilla. Helppojen tehtävien kohdalla eroa ei ollut. Weltyn ja Stemplen mukaan tulokset tukivat heidän hypoteesiaan, eli kielen proseduraalisuudella on vaikutusta vaikeiden kyselyiden onnistumiseen.

QBD

Catarci ja Santucci [1995] vertasivat tutkimuksessaan SQL:n ja QBD-nimisen graafisen kyselykielen käytettävyyttä. Koehenkilöt olivat yliopisto-opiskelijoita, sihteereitä ja asiantuntijoita. Koehenkilöt luokiteltiin kolmeen eri luokkaan tietoteknisen tietämyksen perusteella. Ensimmäisen ryhmän koehenkilöillä (58 henkilöä) ei ollut juurikaan tietämystä tietojenkäsittelystä, toisella ryhmällä (30 henkilöä) oli jonkin verran perustietämystä tietojenkäsittelystä mutta ei tietokannoista,

ja kolmas ryhmä (16 henkilöä) koostui tietojenkäsittelyn asiantuntijoista, joilla oli tietämystä myös tietokannoista. Kaikista kolmesta ryhmästä puolet koehenkilöistä teki kokeen SQL-kielellä ja puolet QBD-kielellä. Ryhmät yritettiin jakaa niin, että molemmissa puolikkaissa olisi esimerkiksi iältään, koulutustaustaltaan, sukupuoleltaan ja työkokemukseltaan mahdollisimman samankaltaisia koehenkilöitä. Ennen koetta SQL-kokeeseen osallistuville koehenkilöille järjestettiin SQL-kurssi ja QBD-kokeeseen osallistuville koehenkilöille QBD-kurssi. Kurssit järjestettiin itseopiskeluna tietokoneella, minkä lisäksi koehenkilöiden oppimisprosessia seurattiin välikokein.

Koetehtävät koostuivat monimutkaisuudeltaan (complexity) kolmen eri tason tehtävistä: matalan tason, keskitason ja korkean tason tehtävät. Jokaiseen näistä kategorioista kuului kuusi tehtävää. Tehtävät olivat luonnollisella kielellä olevia kysymyksiä, joiden pohjalta koehenkilöiden tuli muotoilla kyselykielinen kysely. Tehtävät olivat samoja sekä SQL-kielen kokeessa että QBD-kielen kokeessa. Ensimmäisen ryhmän koehenkilöt tekivät vain matalan tason tehtäväsarjan, toisen ryhmän koehenkilöt matalan ja keskitason tehtäväsarjat ja kolmannen ryhmän koehenkilöt kaikki tehtäväsarjat.

Kokeen tuloksista selvitettiin koehenkilöiden kyselyissä tekemien virheiden määrä sekä kyselyihin käytetty aika. Catarcin ja Santuccin mukaan koehenkilöt menestyivät tehtävissä pääsääntöisesti hieman paremmin QBD-kielellä kuin SQL-kielellä. Koehenkilöt esimerkiksi tekivät erityisesti yksinkertaisissa kyselyissä vähemmän virheitä QBD-kielellä kuin SQL-kielellä.

VisualSQL

Jaakkola ja Thalheim [2003] tutkivat SQL:n ja VisualSQL-nimisen graafisen kyselykielen eroa käyttäjäkokeella. VisualSQL on SQL-kielen graafinen laajennos, eli VisualSQL-kyselyt tehdään määrittelemällä kaavioita, jotka käännetään SQL-kielelle. Tutkimuksen tarkoituksena oli selvittää, eroavatko koehenkilöiden eri kyselykielillä tekemät yksinkertaiset ja monimutkaiset kyselyt käsitteelliseltä oikeellisuudeltaan ja täydellisyydeltään toisistaan.

Koehenkilöt olivat tietojenkäsittelytieteen opiskelijoita, jotka osallistuivat kahdelle kurssille. Toiselle kurssille osallistuneille opiskelijoille opetettiin SQL-kieltä ja toiselle kurssille osallistuneille opiskelijoille VisualSQL-kieltä. Molemmissa ryhmissä oli 24 opiskelijaa ja molempia kieliä opetettiin opiskelijoille sama verran. Molemmille ryhmille järjestettiin tunnin kestävä koetilaisuus, jossa oli sekä yksinkertaisia että monimutkaisia tehtäviä. Tehtävät olivat luonnollisella kielellä annettuja kysymyksiä, joiden pohjalta tuli muotoilla vastauksen kysymykseen antava kyselykielinen kysely. Koehenkilöiden vastaukset tehtäviin arvioitiin välillä 0 (täysin väärin) ja 1 (täysin oikein). Arviointi tehtiin käyttämällä matemaattista kaavaa, joilla koehenkilöiden vastauksia verrattiin oikeisiin ratkaisuihin. Tutkimustuloksen perusteella VisualSQL-kielellä monimutkaisten kyselyiden oikeellisuus on parempi kuin SQL-kielellä.

4. Koeasetelma

Tässä luvussa kuvataan PathSQL- ja SQL-kieliä vertailevan käyttäjäkokeen käytännön järjestelyt ja toteutus. Luvussa esitellään koetilaisuus, koehenkilöt ja koetehtävät, sekä kokeessa kerätty data ja sen käsittely.

4.1. Kokeessa käytetty tietokanta

Koetehtävissä käytettiin tietokantaa, jonka hierarkkinen rakenne vastaa kuvaa 3. Tietokannan rakenne on kuvattu kuvassa 4. Tietokannassa on instituutioita (institute), joilla voi olla osastoja (department), työntekijöitä (employee) ja hallintoelimiä (admin). Osastoilla voi olla projekteja ja työntekijöitä. Lisäksi projekteissa ja hallintoelimissä voi olla työntekijöitä. Kaikissa tietokannan tauluissa on nimi ja avain (id). Lisäksi hierarkian alemmilla tasoilla tauluissa on vierasavaimia, jotka viittaavat välittömästi hierarkiassa yläpuolella oleviin tauluihin.

4.2. Koetehtävät

Koetehtävät koostuivat kahdesta seitsemän tehtävän sarjasta, joissa molemmissa koehenkilöiden tuli muotoilla kyselykielellä oleva kysely luonnollisella kielellä olevan kysymyksen pohjalta. Koetehtävien tyyppi oli siis Reisnerin [1981] luokituksen mukainen kyselyn kirjoitustehtävä. Toisessa sarjassa käytettävä kyselykieli oli PathSQL ja toisessa SQL. Koetehtävien tehtävänannot on kuvattu taulukossa 3. Tehtävät on numeroitu siten, että varsinaisten koetehtävien numerot ovat PathSQL-sarjassa 4-10 ja SQL-sarjassa 12-18 (tehtävänumerot 0-3 ovat koetilaisuudessa tehtyjä harjoitustehtäviä). Esimerkkejä oikean vastauksen tuottavista kyselyistä löytyy taulukosta 4. Taulukon 4 ratkaisut eivät ole ainoita mahdollisia oikeita vastauksia, vaan myös muunlaiset kyselyt voivat tuottaa oikean tuloksen.

Taulukko 3. Koetehtävien tehtävänannot tehtävänumeroineen.

Tehtävänumero	**Tehtävänanto**
PathSQL-sarja: 4 SQL-sarja: 12	Valitse instituutioiden (institute) ja niihin liittyvien osastojen (department) nimet. (Anna attribuutit mainitussa järjestyksessä.)
PathSQL-sarja: 5 SQL-sarja: 13	Valitse instituutioiden ja niihin liittyvien projektien nimet. (Anna attribuutit mainitussa järjestyksessä.)
PathSQL-sarja: 6 SQL-sarja: 14	Valitse projektien nimet ja projekteihin liittyvien osastojen nimet. (Anna attribuutit mainitussa järjestyksessä.)
PathSQL-sarja: 7 SQL-sarja: 15	Valitse instituutioon 'uta' kuuluvien projektien nimet.
PathSQL-sarja: 8 SQL-sarja: 16	Valitse instituutioiden ja niiden aliorganisaatioiden (department, project, admin) työntekijöiden nimet. (Huomaa, että employee-relaatiossa voi olla muitakin työntekijöitä.)
PathSQL-sarja: 9 SQL-sarja: 17	Valitse niiden työntekijöiden nimet, jotka työskentelevät osastolla 'sis' tai sen projektissa.
PathSQL-sarja: 10 SQL-sarja: 18	Valitse instituutioiden nimet ja niihin liittyvien osastojen ja hallintoelimien (admin) nimet. (Anna attribuutit mainitussa järjestyksessä.)

Taulukko 4. Esimerkit koetehtäviin oikean tuloksen tuottavista kyselyistä.

Tehtävä	**Esimerkkivastaus**
4	SELECT institute.name, department.name FROM institute/department
5	SELECT institute.name, project.name FROM institute/department/project
6	SELECT project.name, department.name FROM department/project
7	SELECT project.name FROM institute/department/project WHERE institute.name = 'uta'
8	SELECT employee.name FROM institute//employee
9	SELECT employee.name FROM department//employee WHERE department.name = 'sis'
10	SELECT institute.name, department.name, admin.name FROM institute(/department, /admin)
12	SELECT institute.name, department.name FROM institute, department WHERE institute.id = department.i_id
13	SELECT institute.name, project.name FROM institute, department, project WHERE institute.id = department.i_id AND department.id = project.d_id
14	SELECT project.name, department.name FROM department, project WHERE department.id = project.d_id
15	SELECT project.name FROM institute, department, project WHERE institute.id = department.i_id AND department.id = project.d_id AND institute.name = 'uta'
16	(SELECT employee.name FROM institute, department, project, employee WHERE institute.id = department.i_id AND department.id = project.d_id AND project.id = employee.p_id) UNION (SELECT employee.name FROM institute, admin, employee WHERE institute.id = admin.i_id AND admin.id = employee.a_id) UNION (SELECT employee.name FROM institute, department, employee WHERE institute.id = department.i_id AND department.id = employee.d_id) UNION (SELECT employee.name FROM institute, employee WHERE institute.id = employee.i_id)
17	(SELECT employee.name FROM department, project, employee WHERE department.id = project.d_id AND project.id = employee.p_id AND department.name = 'sis') UNION (SELECT employee.name FROM department, employee WHERE department.id = employee.d_id AND department.name = 'sis')
18	SELECT institute.name, department.name, admin.name FROM institute, department, admin WHERE institute.id = department.i_id AND institute.id = admin.i_id

4.3. Koetilaisuus

Koehenkilöinä käytettiin Tampereen yliopiston tietojenkäsittelyopin aineopintoihin kuuluvan Tietokantaohjelmointi-kurssin kevään 2014 opiskelijoita. Koehenkilöitä oli yhteensä 40. Koetilaisuuksia järjestettiin viisi, ja yksi koehenkilöistä suoritti kokeen kotona omalla tietokoneellaan. Ensimmäiseen koetilaisuuteen osallistui 3, toiseen 5, kolmanteen 16, neljänteen 12 ja viidenteen 3 henkilöä.

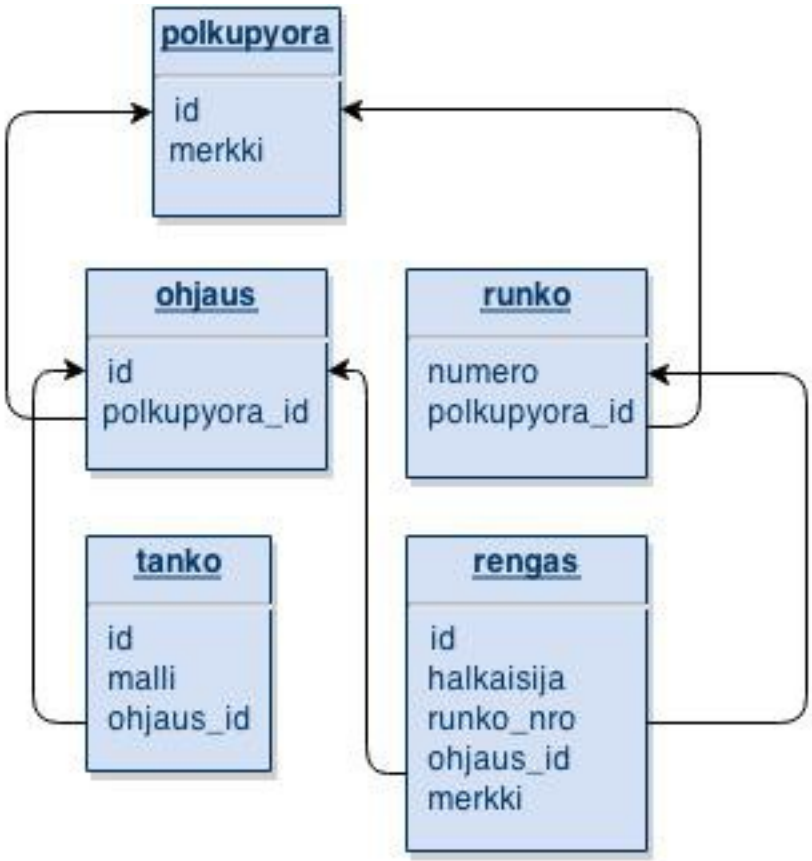
Tietokantaohjelmointi-kurssin luennolla oli esitelty PathSQL-kieli, ja luennolla käsitelty materiaali oli kurssin opiskelijoiden saatavissa verkkosivuilla. Kaikki koehenkilöt eivät kuitenkaan osallistuneet luennolle, ja omatoimista tutustumista materiaaliin ei kontrolloitu mitenkään. Esimerkiksi PathSQL-kieleen liittyviä harjoitustehtäviä ei tehty.

Koetilaisuudet järjestettiin yliopiston tietokoneluokassa, jossa kukin koehenkilö kirjautui koejärjestelmään. Koehenkilöille esiteltiin lyhyesti PathSQL-kieli ja he tekivät ohjatusti neljä harjoitustehtävää koejärjestelmässä PathSQL-kielellä. Harjoitustehtävät muistuttivat varsinaisia koetehtäviä eli niissä koehenkilöiden tuli muodostaa luonnollisella kielellä olevan tehtävänannon perusteella kyselykielinen kysely. Harjoitustehtävien kyselyt kohdistuivat polkupyörien osien muodostamaan rakenteeseen. Esittelyyn ja harjoitustehtävien tekoon käytettiin jokaisessa koetilaisuudessa aikaa kymmenen minuuttia ja esittelyn ja ohjauksen suoritti joka kerralla sama henkilö. Harjoitustehtävät oikeine ratkaisuineen on kuvattu taulukossa 5. Harjoitustehtäviin liittyvä tietokanta ja sen pohjana oleva polkupyörän osien muodostama hierarkia on kuvattu kuvissa 5 ja 6.

Taulukko 5. Neljä PathSQL-harjoitustehtävää oikean tuloksen tuottavine esimerkkivastauksineen.

Tehtävänanto	**Esimerkkivastaus**
Valitse polkupyörien merkit ja niihin liittyvät runkojen numerot.	SELECT polkupyora.merkki, runko.numero FROM polkupyora/runko
Valitse merkkiä 'jopo' olevien polkupyörien runkojen numerot.	SELECT runko.numero FROM polkupyora/runko WHERE polkupyora.merkki = 'jopo'
Valitse merkkiä 'helkama' olevien polkupyörien renkaiden halkaisijat.	SELECT rengas.halkaisija FROM polkupyora//rengas WHERE polkupyora.merkki = 'helkama'
Mitä ohjaustangon mallia käytetään missäkin rungossa?	SELECT tanko.malli, runko.numero FROM polkupyora(/ohjaus/tanko,/runko)

Kuva 5.Polkupyörän osien muodostama hierarkia.



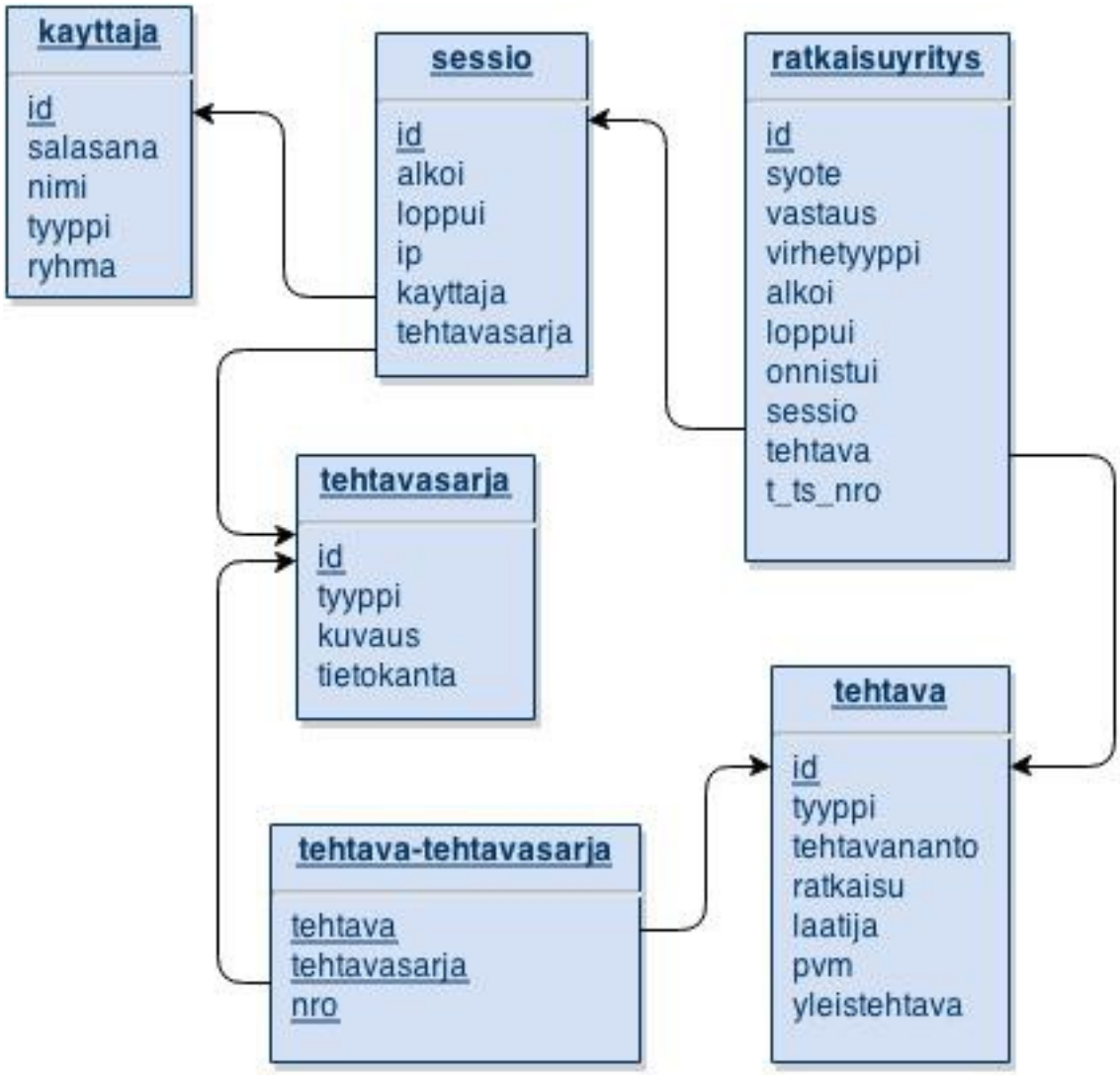
Kuva 6. Harjoitustehtävien tietokanta, joka perustuu kuvan 5 hierarkiaan.

Harjoitustehtävien jälkeen koehenkilöt suorittivat itsenäisesti koejärjestelmässä taulukossa 3 kuvatut koetehtävät. Kokeen aikana PathSQL-harjoitustehtäviin perustuva esimerkkimateriaali oli koehenkilöiden nähtävissä (katso liite 1), kuten myös kuvan 3 hierarkia ja siihen perustuva kuvan 4 tietokantakaavio. Koe oli tyypiltään siis loppukoe [Reisner, 1981], mutta koska polkuilmaisujen käyttö voidaan nähdä yhtenä uutena SQL-kielen funktiona, koe voidaan nähdä myös Reisnerin luokituksen mukaan välittömän ymmärryksen testaamisena. Kokeessa tutkittiin enemmänkin polkuilmaisujen soveltamista kuin niiden ulkoaoppimista. Koehenkilöistä 16 teki SQL-tehtäväsarjan ja 24 PathSQL-tehtäväsarjan ensin. Tehtävien suoritukseen käytettävää aikaa ei rajattu, eli koehenkilöt saivat poistua koetilaisuudesta suoritettuaan tehtävät ja käyttää tehtävien tekoon niin paljon aikaa kuin tahtoivat.

Kokeessa käytetyssä järjestelmässä annettiin valmiina kyselyn alkuosa, eli koko SELECT-osa FROM-osaan asti. Esimerkiksi kyselyssä 4 annettu alkuosa oli SELECT institute.name, department.name, jonka perään koehenkilö kirjoitti kyselyn loppuosan. Järjestelmä vertasi koehenkilön antaman syötteen tuottamaa tulostaulua oikean vastauksen tuottamaan tulostauluun ja antoi koehenkilölle tiedon siitä oliko kysely oikein vai väärin. Jos kyselystä puuttui SELECT- tai FROM-osa, järjestelmä antoi tästä ilmoituksen. Järjestelmä ei antanut koehenkilöille muuta palautetta. Jos vastaus oli väärin, järjestelmä palautti sen ja koehenkilö saattoi jatkaa seuraavassa ratkaisuyrityksessä sen pohjalta. Yhteen tehtävään sai käyttää enintään neljä ratkaisuyritystä.

4.4. Koedatan keräys ja käsittely

Koejärjestelmä tallensi tiedot koehenkilöiden suorituksista tietokantaan, jonka rakenne on kuvattu kuvassa 7. Kunkin koehenkilön tunnus (id) ja salasana oli tallennettu ennen koetta tietokantaan (kuvan 7 taulu kayttaja), ja koehenkilöt käyttivät näitä kirjautuessaan järjestelmään. Myöhemmin tunnuksia käytettiin kerätyn datan analysoinnissa yksilöimään koehenkilöitä. Sessio-tauluun tallennettiin tiedot koesessioista, eli koehenkilön id, ip-osoite, tehtäväsarjan suorituksen aloitus- ja lopetusajat sekä suoritetun tehtäväsarjan id. Jokainen koehenkilön tekemä ratkaisuyritys tallennettiin ratkaisuyritys-tauluun. Tallennettuja tietoja olivat esimerkiksi tehtävän id, yrityksen aloitus- ja lopetusaika, session id, koehenkilön antama syöte sekä tieto siitä, onnistuiko ratkaisuyritys vai ei. Lisäksi tietokannassa on oma taulu tehtäville (tehtava) ja tehtäväsarjoille, joihin tehtävät kuuluvat (tehtavasarja). Taulu tehtava-tehtavasarja liittää tehtävät tehtäväsarjoihinsa. Harjoitustehtäville, SQL- ja PathSQL-tehtäville on kullekin oma tehtäväsarjansa.



Kuva 7. Koejärjestelmän tietokannan rakenne.

Tietokantaan tallentuneista tiedoista tehtiin hakuja SQL-kyselykielellä (esimerkkejä kyselyistä liitteessä 2) ja tulokset tallennettiin analysointia varten laskentataulukoihin. Haettavia tietoja olivat esimerkiksi tehtävän onnistuneen vastauksen ratkaisuyrityskerran pienin, suurin ja keskimääräinen arvo, sekä pienin, suurin ja keskimääräinen suoritusaika onnistuneessa tai epäonnistuneessa vastauksessa. Suoritusajat muunnettiin kyselyissä SQL:n aikaleimoista sekunneiksi tilastollisen analysoinnin ja visualisoinnin helpottamiseksi. Tulosten analyysoinnissa ja visualisoinnissa käytettiin SPSS-tilasto-ohjelmaa. Lisäksi suoritusaikojen rinnakkaiskoordinaattikuvien teossa käytettiin Matlab-ohjelmaa. Koehenkilöille annettiin uudet, alkuperäisistä poikkeavat tunnistenumerot (id) kuvissa ja tekstissä tapahtuvaa yksilöintiä varten. Tämä tehtiin yksityisyyden suojan varmistamiseksi, sillä kokeen tulosten yhteydessä käsitellään esimerkiksi koehenkilöiden

kurssiarvosanoja. Osa koehenkilöistä tai heidän ratkaisuyrityksistään jätettiin pois joko kaikista tai osasta analyyseja ja kuvia, sillä he olivat esimerkiksi yrittäneet ratkaista tehtäviä väärällä kielellä tai käyttäneet liian monta (yli neljä) ratkaisuyrityskertaa.

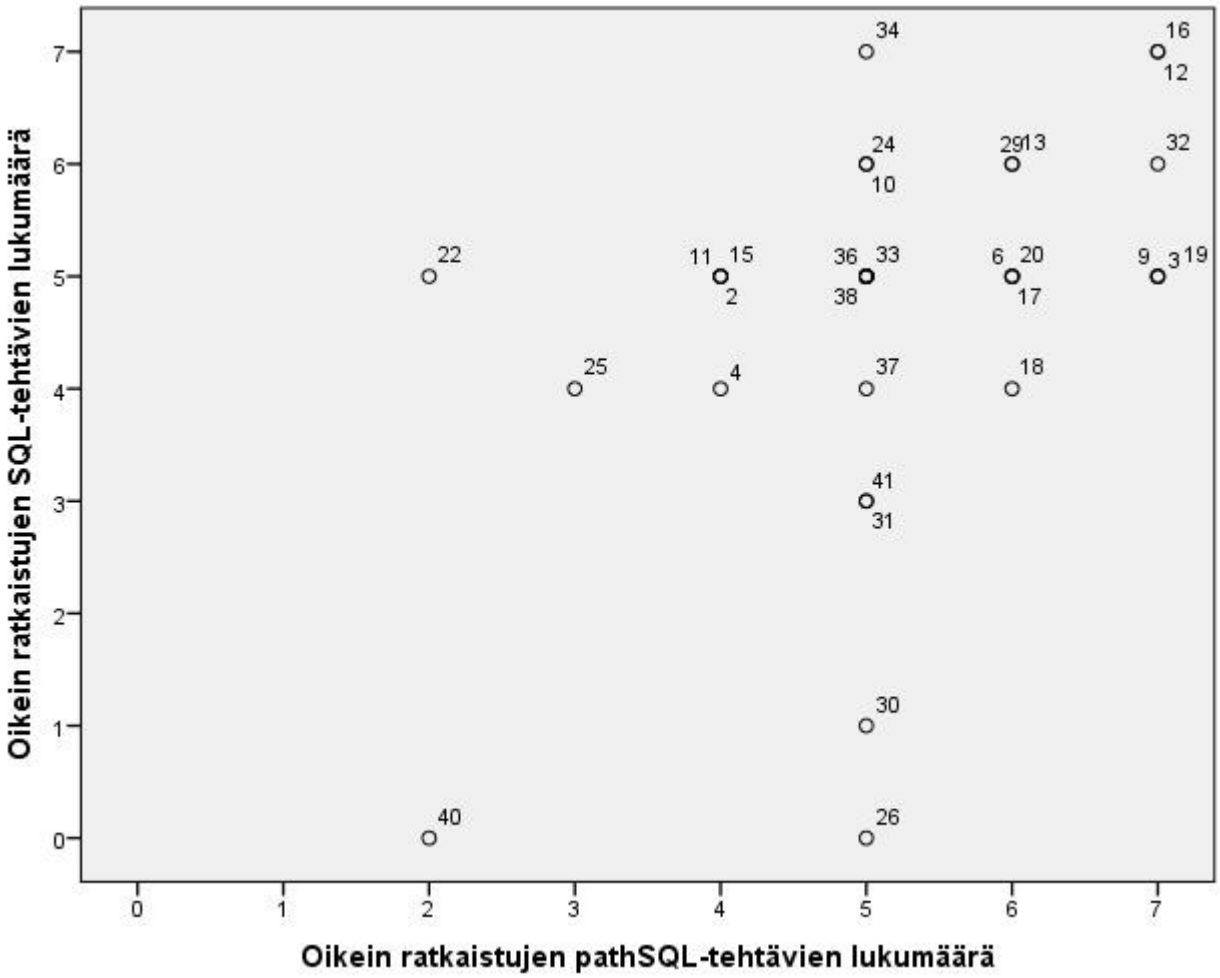
5. Kokeen tulokset

Tässä luvussa esitellään edellisessä luvussa kuvatun kokeen tulokset. Ensin tarkastellaan sitä, onnistuivatko koehenkilöt paremmin PathSQL- vai SQL-tehtävien ratkaisemisessa, ja oliko aiemmalla opintomenestyksellä yhteyttä onnistumiseen. Koehenkilöiden suorituksista etsitään myös mahdollisia yhteyksiä tehtävissä epäonnistumisten välillä käyttäen apuna kattavien joukkojen tiedonlouhintaa. Tämän jälkeen vertaillaan tehtävien ratkaisuun käytettyä aikaa.

5.1. Oikein ratkaistut tehtävät

Oikein ratkaistujen PathSQL- ja SQL-tehtävien määrien kokonaisvertailussa oli mukana 34 koehenkilöä. Oikein mennyt suoritus tarkoittaa sitä, että koehenkilö on ratkaissut tehtävän oikein viimeistään neljännellä eli viimeisellä ratkaisuyrityskerralla. Kuusi koehenkilöä jätettiin pois vertailusta, sillä vertailu ei olisi ollut näissä tapauksissa mielekästä. Kaksi poisjätetyistä koehenkilöistä oli tehnyt SQL-tehtäväsarjan kokonaan ja kaksi osittain PathSQL-kielellä. Lisäksi yksi oli jättänyt PathSQL-sarjan kokonaan tekemättä ja yhdellä oli osassa tehtävistä liikaa ratkaisuyrityksiä. Kolme näistä poisjätetyistä koehenkilöstä on mukana osassa tehtäväkohtaisia tarkasteluja.

34:stä koehenkilöistä 16:lla oli enemmän oikein menneitä tehtäviä PathSQL-sarjassa, 8:lla SQL-sarjassa ja 10:llä koehenkilöllä oli saman verran oikein menneitä tehtäviä kummassakin sarjassa. Koehenkilöt olivat ratkaisseet oikein keskimäärin 5,18 PathSQL-tehtävää ja 4,68 SQL-tehtävää. Koehenkilöt vaikuttaisivat siis tämän perusteella ratkaisevan tehtävät paremmin PathSQL-kielellä. Eroa PathSQL- ja SQL-tehtäväsarjojen välillä testattiin Wilcoxonin merkittyjen sijalukujen testillä [Wilcoxon, 1945], ja p-arvoksi saatiin 0,068. Ero ei ole tilastollisesti merkitsevä, eli koehenkilöt ratkaisivat tehtävät kummallakin kielellä suunnilleen yhtä hyvin.



Kuva 8. Koehenkilöiden oikein ratkaisemien PathSQL- ja SQL-tehtävien lukumäärät (n=34).

Kuvassa 8 on kuvattuna yksittäisten koehenkilöiden oikein ratkaistujen PathSQL- ja SQL-tehtävien lukumäärät. Kuvasta voidaan nähdä, että useimmat niistä koehenkilöstä, jotka ovat onnistuneet ratkaisemaan useita tehtäviä onnistuneesti toisella kielistä, ovat menestyneet hyvin myös toisen kielen tehtävissä. Lisäksi kuvasta voidaan havaita, että enemmistö koehenkilöistä selviytyi kokeesta melko hyvin. Huomionarvoista on myös se, että kaksi koehenkilöistä ei onnistunut ratkaisemaan oikein yhtään SQL-sarjan tehtävää, mutta onnistui kuitenkin joissakin PathSQL-sarjan tehtävissä. Näistä toinen, koehenkilö 26, onnistui jopa viidessä PathSQL-sarjan tehtävässä. Lisäksi koehenkilö 30 onnistui suorittamaan onnistuneesti vain yhden SQL-sarjan tehtävän, mutta vastasi oikein viidessä PathSQL-sarjan tehtävässä. Koehenkilö 40 ei saanut myöskään yhtään tehtävää oikein SQL-sarjassa mutta onnistui kahdessa PathSQL-tehtävässä. Toisin päin vastaavaa ei ole juurikaan havaittavissa, eli SQL-tehtävissä menestyneet koehenkilöt

menestyivät pääsääntöisesti hyvin myös PathSQL-tehtävissä. Poikkeuksena on koehenkilö 22, joka menestyi SQL-tehtävissä melko hyvin, mutta sai ainoastaan kaksi PathSQL-tehtävää oikein.

Taulukko 6. Oikein ratkaistujen PathSQL- ja SQL-tehtävien lukumäärät ja osuudet.

PathSQL-tehtävä	**n**	**Oikein ratkaistujen lukumäärä**	**Oikein ratkaistujen osuus (%)**	**SQL-tehtävä**	**n**	**Oikein ratkaistujen lukumäärä**	**Oikein ratkaistujen osuus (%)**
4	34	33	97	12	34	30	88
5	35	33	94	13	35	31	89
6	35	34	97	14	35	31	89
7	36	32	89	15	36	32	89
8	37	10	27	16	37	7	19
9	37	9	24	17	37	8	22
10	37	33	89	18	37	27	73

Koehenkilöiden onnistumista tehtävissä tarkasteltiin myös tehtäväparikohtaisesti. Taulukossa 6 on onnistuneiden suoritusten osuudet kaikista tarkastelussa mukana olleiden PathSQL- ja SQL-tehtävien suorituksista. Tehtävänannoltaan vastaavat PathSQL- ja SQL-tehtävät ovat taulukossa vierekkäin. Taulukosta voidaan nähdä, että koehenkilöt ovat onnistuneet ratkaisemaan kaikkien tehtäväparien kohdalla suuremman osan tehtävistä oikein PathSQL-kielellä kuin SQL-kielellä lukuun ottamatta tehtävää 7, joissa onnistuneiden suoritusten osuudet ovat samat.

Taulukko 7. Niiden koehenkilöiden määrä, jotka ovat ratkaisseet tehtäväparista 4 ja 12 oikein molemmat, vain toisen tai ei kumpaakaan tehtävää (n=34).

	Oikein ratkaistut PathSQL	**Väärin ratkaistut PathSQL**
Oikein ratkaistut SQL	29	1
Väärin ratkaistut SQL	4	0

Taulukko 8. Niiden koehenkilöiden määrä, jotka ovat ratkaisseet tehtäväparista 5 ja 13 oikein molemmat, vain toisen tai ei kumpaakaan tehtävää (n=35).

	Oikein ratkaistut PathSQL	**Väärin ratkaistut PathSQL**
Oikein ratkaistut SQL	30	1
Väärin ratkaistut SQL	3	1

Taulukko 9. Niiden koehenkilöiden määrä, jotka ovat ratkaisseet tehtäväparista 6 ja 14 oikein molemmat, vain toisen tai ei kumpaakaan tehtävää (n=35).

	Oikein ratkaistut PathSQL	**Väärin ratkaistut PathSQL**
Oikein ratkaistut SQL	30	1
Väärin ratkaistut SQL	4	0

Taulukko 10. Niiden koehenkilöiden määrä, jotka ovat ratkaisseet tehtäväparista 7 ja 15 oikein molemmat, vain toisen tai ei kumpaakaan tehtävää (n=36).

	Oikein ratkaistut PathSQL	**Väärin ratkaistut PathSQL**
Oikein ratkaistut SQL	29	3
Väärin ratkaistut SQL	3	1

Taulukko 11. Niiden koehenkilöiden määrä, jotka ovat ratkaisseet tehtäväparista 8 ja 16 oikein molemmat, vain toisen tai ei kumpaakaan tehtävää (n=37).

	Oikein ratkaistut PathSQL	**Väärin ratkaistut PathSQL**
Oikein ratkaistut SQL	4	3
Väärin ratkaistut SQL	6	24

Taulukko 12. Niiden koehenkilöiden määrä, jotka ovat ratkaisseet tehtäväparista 9 ja 17 oikein molemmat, vain toisen tai ei kumpaakaan tehtävää (n=37).

	Oikein ratkaistut PathSQL	**Väärin ratkaistut PathSQL**
Oikein ratkaistut SQL	4	4
Väärin ratkaistut SQL	5	24

Taulukko 13. Niiden koehenkilöiden määrä, jotka ovat ratkaisseet tehtäväparista 10 ja 18 oikein molemmat, vain toisen tai ei kumpaakaan tehtävää (n=37).

	Oikein ratkaistut PathSQL	**Väärin ratkaistut PathSQL**
Oikein ratkaistut SQL	25	2
Väärin ratkaistut SQL	8	2

Taulukoista 7-13 voidaan nähdä tehtäväparikohtaisesti, kuinka moni koehenkilöistä onnistui molemmissa, vain toisessa tai ei kummassakaan toisiaan vastaavista PathSQL- ja SQL-tehtävistä. Useimmissa tehtäväpareissa enemmistö koehenkilöistä menestyi hyvin molemmissa tehtävissä. Poikkeuksena olivat tehtäväparit 8 ja 16 sekä 9 ja 17, joissa enemmistö koehenkilöistä epäonnistui molemmissa tehtävissä. Myös näistä taulukoista voidaan nähdä se, että tyypillisemmin koehenkilöt epäonnistuivat SQL-tehtävissä ja onnistuivat PathSQL-tehtävissä kuin toisin päin tai niin etteivät he onnistuneet kummassakaan tehtävässä.

Taulukossa 14 on listattu kuinka monta ratkaisuyritystä koehenkilöt ovat käyttäneet vähintään, enintään ja keskimäärin ennen kuin tehtävän ratkaisu on onnistunut, sekä onnistuneesti ratkaistujen tehtävien ratkaisuyrityskertojen mediaani. Taulukon perusteella voidaan havaita, että PathSQL-tehtävien (tehtävät 4-10) oikeisiin ratkaisuihin on vaadittu keskimäärin hieman vähemmän ratkaisuyrityksiä kuin SQL-tehtävien (tehtävät 12-18), lukuun ottamatta tehtäviä 6 ja 14, 8 ja 16 sekä 9 ja 17. Tehtävistä helpoimmat näyttävät olevan tehtävät 5 ja 7, joissa kaikki tehtävän oikein ratkaisseet koehenkilöt ovat onnistuneet viimeistään toisella ratkaisuyrityskerralla. Hankalin tehtävistä vaikuttaa olevan tehtävä 9.

Taulukko 14. Oikeaan vastaukseen käytetty ratkaisuyrityskertojen vähimmäis- ja enimmäismäärä, keskiarvo sekä mediaani PathSQL- ja SQL-tehtävissä.

Tehtävä	n	Vähintään	Enintään	Keski-arvo	Me-diaani	Tehtävä	n	Vähintään	Enintään	Keski-arvo	Me-diaani
4	33	1	4	1,24	1	12	30	1	4	1,37	1
5	33	1	2	1,03	1	13	31	1	3	1,35	1
6	34	1	4	1,38	1	14	31	1	3	1,16	1
7	32	1	2	1,25	1	15	32	1	3	1,44	1
8	10	1	4	1,8	1	16	7	1	3	1,57	1
9	9	1	4	2,33	3	17	8	1	3	1,75	1,5
10	33	1	3	1,24	1	18	27	1	4	1,41	1

Taulukossa 15 on kuvattuna erotukset SQL- ja PathSQL-tehtävissä oikeaan ratkaisuun vaadittujen ratkaisuyrityskertojen välillä (SQL-PathSQL). Mukana tarkastelussa ovat ainoastaan ne koehenkilöt, jotka ratkaisivat molemmissa tehtäväsarjoissa toisiaan vastaavat tehtävät oikein. Taulukossa on erotusten vähimmäis- ja enimmäisarvot sekä keskiarvo ja mediaani. Tehtävissä 8 ja 16 sekä 9 ja 17 koehenkilöitä, jotka onnistuivat ratkaisemaan molemmat tehtävät, oli hyvin vähän. Muissa tehtävissä tällaisia koehenkilöitä oli enemmistö koehenkilöistä eli huomattavasti enemmän. Lukuun ottamatta tehtäväparia 8 ja 16 kaikkien tehtäväparien kohdalla mediaani oli 0, mutta keskiarvo jäi useimmilla tehtäväpareilla hieman tämän yläpuolelle. Tehtäväparin 8 ja 16 lisäksi myös tehtäväparissa 6 ja 14 koehenkilöt käyttivät vähemmän ratkaisuyrityskertoja SQL- kuin PathSQL-tehtävän ratkaisuun. PathSQL-tehtävät vaativat siis keskimäärin hiukan vähemmän ratkaisuyrityksiä oikean vastauksen saamiseksi. Tehtäväpareissa 6 ja 14 sekä 8 ja 16 PathSQL-tehtävässä ratkaisuyrityksiä tarvittiin puolestaan hieman enemmän kuin SQL-tehtävässä. Kaiken kaikkiaan erot ovat hyvin pieniä, eli molemmilla kielillä ratkaisuyrityksiä vaadittiin suunnilleen yhtä paljon.

Taulukko 15. Oikeaan ratkaisuun vaadittujen ratkaisuyrityskertojen määrän erotuksen (SQL - PathSQL) vähimmäis- ja enimmäismäärä sekä keskiarvo ja mediaani tehtäväpareittain.

Tehtävät	**Koehenkilöiden lukumäärä**	**Erotus ratkaisuyrityskertojen välillä**			
		Vähintään	**Enintään**	**Keskiarvo**	**Mediaani**
4 ja 12	29	-3	3	0,14	0
5 ja 13	30	-1	2	0,33	0
6 ja 14	30	-3	1	-0,2	0
7 ja 15	29	-1	2	0,14	0
8 ja 16	4	-3	1	-1	-1
9 ja 17	4	-1	1	0	0
10 ja 18	25	-2	3	0,2	0

Ensimmäisessä koetilaisuudessa koehenkilöille annetussa tietokantakaaviossa oli virhe: relaatioiden viiteavaimet oli merkitty väärin. Koetilaisuuteen osallistui yhteensä kolme henkilöä: koehenkilöt 3, 17 ja 21. Koehenkilö 3 onnistui ratkaisemaan kaikki PathSQL-sarjan tehtävät oikein. SQL-tehtäväsarjassa koehenkilö ratkaisi viisi tehtävää oikein ja kaksi tehtävää (tehtävät 16 ja 17) kaikilla ratkaisuyrityskerroilla väärin. Koehenkilö 17 ratkaisi oikein muut PathSQL-sarjan tehtävät paitsi tehtävän 8. SQL-sarjassa hän ratkaisi kaikki tehtävät oikein lukuun ottamatta tehtäviä 16 ja 17. Koehenkilö 21 teki PathSQL-sarjan tehtävistä viisi oikein ja kaksi (tehtävät 8 ja 9) väärin. SQL-sarjan tehtävistä hän teki myös viisi oikein ja kaksi väärin. Nämä kaksi tehtävää (16 ja 17) vastasivat tehtävänannoltaan PathSQL-sarjan tehtäviä 8 ja 9. Verrattuna muihin koetilaisuuksiin osallistuneiden koehenkilöiden suorituksiin koehenkilöt 3, 17 ja 21 eivät tehneet enemmän virheitä.

5.2. Kurssimenestyksen yhteys kokeen tulokseen

Koehenkilöiden oikein menneiden PathSQL- ja SQL-koetehtävien lukumäärää verrattiin heidän opintomenestykseensä kahdella tietojenkäsittelyopin kurssilla. Toisin sanoen haluttiin tietää, saivatko kursseilla hyvin menestyneet opiskelijat enemmän koetehtäviä oikein kuin huonommin menestyneet opiskelijat. Toinen kursseista oli tietojenkäsittelytieteiden perusopintoihin kuuluva Tietokantojen perusteet (TKP), jossa opiskellaan SQL-kyselykielen, SQL-tietokantojen ja ER-mallinnuksen perusteita. Kurssin laajuus on viisi opintopistettä. Toinen kursseista on tietojenkäsittelyopin aineopintoihin kuuluva Tietokantaohjelmointi (TIKO), jossa opiskellaan SQL-kielen ja tietokantojen kehittyneempiä käyttötapoja. Lisäksi kurssilla opiskellaan muun muassa SQL-kielen ja tietokantojen käyttöä osana ohjelmien tekoa, sekä relaatiotietomallin teoreettisia perusteita ja ER-mallinnusta. Kurssin laajuus on kymmenen opintopistettä. Molemmilla kursseilla on käytössä arvosana-asteikko 1-5. Vertailussa käytettiin Spearmanin järjestyskorrelaatiokerrointa.

36:n koehenkilön TKP-kurssin ja 39:n koehenkilön TIKO-kurssin arvosana oli saatavilla. Neljän koehenkilön TKP:n arvosana puuttui. Tämä johtuu todennäköisesti siitä, että nämä opiskelijat suorittavat vain maisterin tutkintoa, eikä heidän ole tarvinnut suorittaa TKP-kurssia. TIKO-kurssin arvosana puuttui yhdeltä koehenkilöltä, jolle ei ole vielä annettu arvosanaa TIKO-kurssista.

TKP:n arvosanojen ja koetehtävissä onnistumisen vertailusta jätettiin pois ne koehenkilöt, joilta ei ollut saatavissa TKP:n arvosanaa. TIKO:n arvosanojen ja koetehtävissä onnistumisen vertailusta jätettiin vastaavasti pois koehenkilö, jolla ei ollut saatavissa TIKO:n arvosanaa. Lisäksi molemmista vertailuista jätettiin pois viisi koehenkilöä, joiden onnistuneiden koetehtävien lukumäärän laskeminen oli ongelmallista: he olivat tehneet SQL-sarjan tehtävät kokonaan tai osittain PathSQL-kielellä tai heillä oli liikaa (yli neljä) ratkaisuyrityksiä.

TKP-kurssin arvosanan sekä oikein ratkaistujen PathSQL- ja SQL-tehtävien lukumäärien vertailussa oli mukana 30 koehenkilöä. Oikein ratkaistujen PathSQL-tehtävien ja TKP:n arvosanan välinen korrelaatio on 0,5, eli kurssilla menestyneet koehenkilöt onnistuivat paremmin myös PathSQL-tehtävissä. TKP:n arvosanan ja oikein ratkaistujen SQL-tehtävien välillä on myös samankaltainen ja lähes yhtä voimakas yhteys: korrelaatio on 0,469.

TIKO-kurssin arvosanan sekä oikein ratkaistujen PathSQL- ja SQL-tehtävien lukumäärien vertailussa oli mukana 33 koehenkilöä. TIKO:n arvosanan yhteys PathSQL-tehtävissä onnistumiseen ei ollut yhtä voimakas kuin TKP:n arvosanan: korrelaatio on 0,163. TIKO:n ja SQL-tehtävissä onnistumisen välinen positiivinen korrelaatio on hieman voimakkaampi (0,307). SQL-kielen osuus TIKO-kurssilla käsiteltävistä asioista on vähäisempi kuin TKP-kurssilla, joten heikompi korrelaatio ei ole yllättävää.

Molempien kurssien arvosanoista laskettiin myös painotettu keskiarvo, jossa painotus tehtiin opintopistemäärien perusteella. Tätä keskiarvoa verrattiin oikein menneiden tehtävien lukumääriin. Vertailussa oli mukana 29 koehenkilöä. TIKO-kurssin arvosanan merkitys oli siis suurempi. Painotetun keskiarvon ja oikein ratkaistujen PathSQL-tehtävien lukumäärän välinen korrelaatio on 0,319. Oikein ratkaistujen SQL-tehtävien lukumäärän ja painotetun keskiarvon välinen korrelaatio on puolestaan 0,344, eli eroa SQL-tehtävien ja PathSQL-tehtävien välillä ei juurikaan ole. Molemmilla on melko lievä positiivinen yhteys kurssiarvosanojen painotettuun keskiarvoon.

Koehenkilö 26 ei ratkaissut oikein yhtään SQL-tehtävää, mutta menestyi melko hyvin PathSQL-tehtävissä (viisi tehtävää oikein seitsemästä). SQL-tehtävien ratkaisuyrityksissä koehenkilö 26 epäonnistui viiteavainten ja liitosten käytössä. Epäonnistumiseen saattaa vaikuttaa ainakin jossain määrin se, ettei koehenkilö 26 ole suorittanut TKP-kurssia, ja TIKO-kurssin arvosana on 2 (tyydyttävä). Myös koehenkilö 30 onnistui PathSQL-tehtävissä huomattavasti paremmin kuin SQL-tehtävissä, mutta heikko menestys SQL-tehtävissä johtui ainakin osittain koehenkilöstä itsestään riippumattomasta syystä: tehtävän vastaukseen oli annettu valmis alkuosa FROM-sanaan asti, ja

koehenkilö käytti kyselyn loppuosassa aliaksia, joita hän ei ollut lisännyt kyselyn valmiiseen alkuosaan. Koehenkilö 30 on saanut molemmilta kursseilta melko matalan arvosanan (2). Koehenkilö 40 ei onnistunut ratkaisemaan yhtään SQL-tehtävää oikein, mutta onnistui kuitenkin kahdessa PathSQL-tehtävässä. Hänen molempien kurssien arvosanansa on 1 (välttävä). SQL-tehtävissä epäonnistuminen johtui osittain siitä, että koehenkilö kirjoitti institute-taulun nimen väärin kahden tehtävän ratkaisuyrityksissä. Lisäksi koehenkilö teki virheitä myös liitosoperaatioissa ja osa ratkaisuyrityksistä oli puutteellisia. Koehenkilö 22 ratkaisi 5 SQL-tehtävää mutta ainoastaan kaksi PathSQL-tehtävää (tehtävät 7 ja 10) oikein. Hän oli saanut molemmilta kursseilta arvosanan 3 (hyvä). Suoritusten epäonnistuminen PathSQL-tehtävissä johtui virheistä polkujen muodostamisessa: koehenkilö ei käyttänyt esimerkiksi //-ilmaisua, ja käytti sulkumerkkejä sellaisissa kyselyissä, joissa niitä ei olisi pitänyt käyttää.

5.3. Epäonnistumisten yhteisesiintymät tehtävittäin

Koehenkilöiden suorituksista etsittiin tiedonlouhinnan avulla mahdollisia kattavia joukkoja koehenkilöiden epäonnistumisissa tehtävien suoritusten joukossa. Toisin sanoen haluttiin tietää, oliko joissain tehtävissä jonkinlainen yhteys epäonnistumisien välillä eli esiintyivätkö joissain tehtävissä epäonnistumiset usein yhdessä. Kattava joukko (frequent itemset) on sellainen alijoukko tietoalkioita jossakin tietoalkiojoukossa, joka esiintyy yhdessä datan riveillä (tai tapauksissa) yhtä usein tai useammin kuin jokin käyttäjän määrittelemä raja-arvo [Han et al., 2007].

Kattavien joukkojen louhinnassa käytettiin Magnum Opus -nimistä ohjelmistoa. Kattavien joukkojen kriteereinä käytettiin kattavuutta (coverage) ja nostovoimaa (leverage). Joukon kattavuus (coverage) tarkoittaa sitä, miten suuressa osassa koehenkilöiden suorituksia on jokin tietty epäonnistuneiden tehtävien joukko [Webb, 2012]. Joukon nostovoima tarkoittaa alkioiden lukumäärää yli suurimman kattavuuden odotusarvon [Webb, 2010]. Tuloksista seulottiin pois tilastollisesti merkityksettömät (insignificant) tulokset, joiden p-arvo on yli 0,05. Mukana louhinnassa oli 34:n koehenkilön suoritukset. Suorituksista jätettiin pois ne, joilla ei aiemmin luvussa 5.1. mainituista syistä voinut laskea onnistumisten kokonaismäärää. Louhinnan tulokset esitetään taulukossa 16.

Taulukko 16. Louhinnassa löytyneet kattavat joukot.

Epäonnistuneet tehtävät	Kattavuus (n)	Nostovoima (n)
16 ja 17	0,677 (23)	0,0692 (2,4)
8 ja 9	0,618 (21)	0,0986 (3,4)
8, 16 ja 17	0,559 (19)	0,0813 (2,8)
8, 9, 16 ja 17	0,5 (17)	0,0796 (2,7)
13, 14 ja 15	0,088 (3)	0,0779 (2,6)
12, 14 ja 15	0,088 (3)	0,0779 (2,6)
12, 13 ja 15	0,088 (3)	0,0779 (2,6)
12, 13 ja 14	0,088 (3)	0,0779 (2,6)
14 ja 15	0,088 (3)	0,0744 (2,5)
13 ja 15	0,088 (3)	0,0744 (2,5)
12 ja 15	0,088 (3)	0,0744 (2,5)
13 ja 14	0,088 (3)	0,0744 (2,5)
12 ja 14	0,088 (3)	0,0744 (2,5)
12 ja 13	0,088 (3)	0,0744 (2,5)
4 ja 6	0,029 (1)	0,0285 (1)

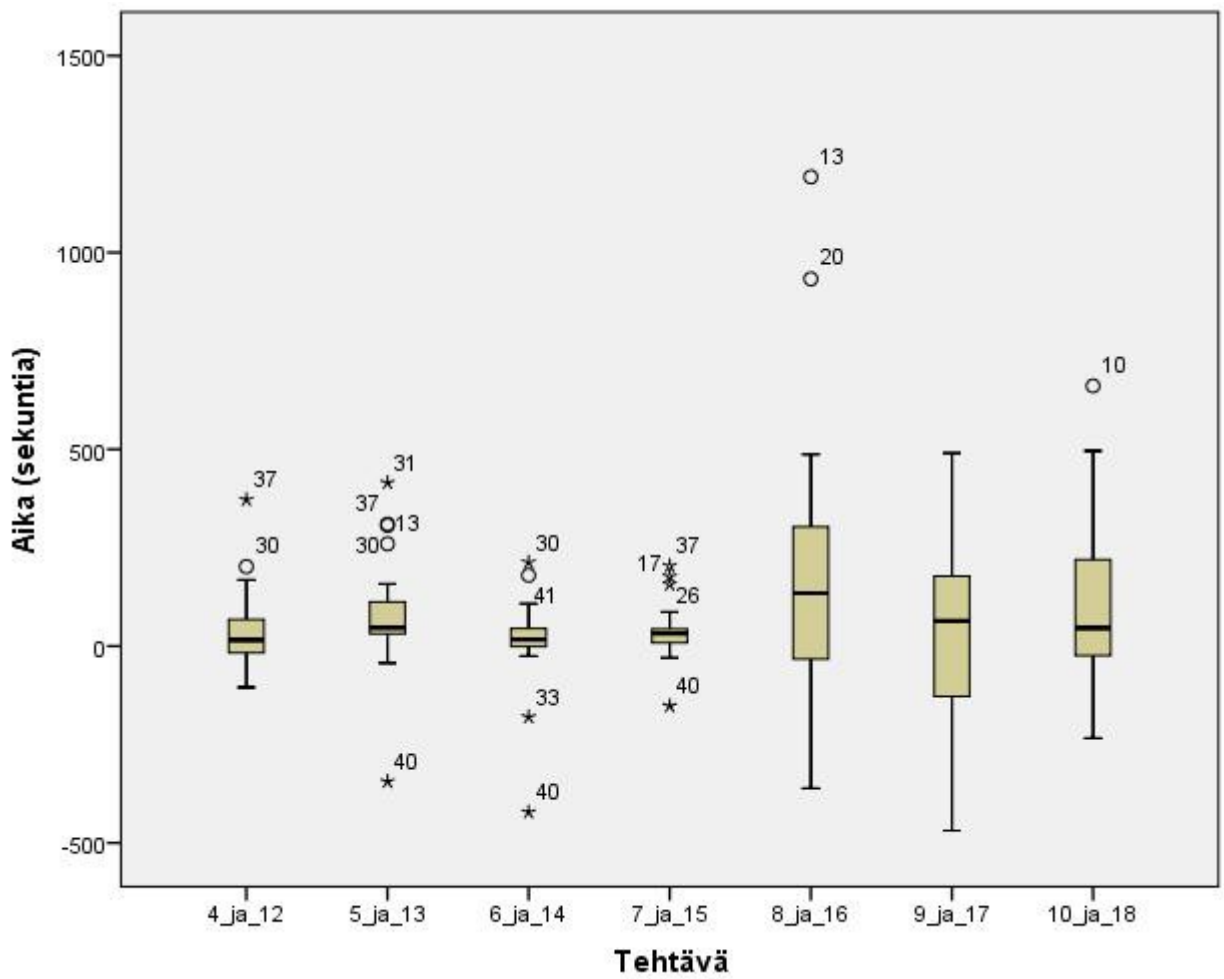
Taulukosta 16 voidaan havaita, että suurin kattavuus oli PathSQL-tehtäväparilla 8 ja 9 sekä SQL-tehtäväparilla 16 ja 17. Yli puolet koehenkilöistä oli epäonnistunut kummassakin näiden tehtäväparien tehtävistä. Yli puolet koehenkilöistä oli epäonnistunut SQL-tehtävien 16 ja 17 lisäksi myös PathSQL-tehtävässä 8. Puolet koehenkilöistä oli epäonnistunut kahdessa tehtävänannoltaan toisiaan vastaavassa tehtäväparissa: 8 ja 16 sekä 9 ja 17. Kaikki neljä edellä mainittua tehtävää olivat myös aiempien tarkasteluiden perusteella vaikeita koehenkilölle, joten niiden esiintyminen myös tässä yhteydessä ei ole yllättävää. Tehtävät ovat myös keskenään melko samankaltaisia: koehenkilön tulee niistä jokaisessa löytää työntekijät, jotka voivat sijaita eri kohdissa hierarkiaa, mikä saattaa hankaloittaa kyselyn muotoilua. Tämän vuoksi se, että samat koehenkilöt epäonnistuvat useammassa niistä, ei ole yllättävää. Vahvin yhteys vaikuttaisi olevan sekä kattavuus

että nostovoima huomioiden tehtävien 8 ja 9 välillä. Näistä molemmat edellyttivät samankaltaista //-operaattorin käyttöä kaikkien employee-relaatioiden saavuttamiseksi.

Yllä mainitun lisäksi tuloksesta voidaan havaita, että kolme koehenkilöä epäonnistui kaikissa SQL-tehtävissä 12, 13, 14 ja 15, mutta eivät kaikissa vastaavissa PathSQL-tehtävissä.

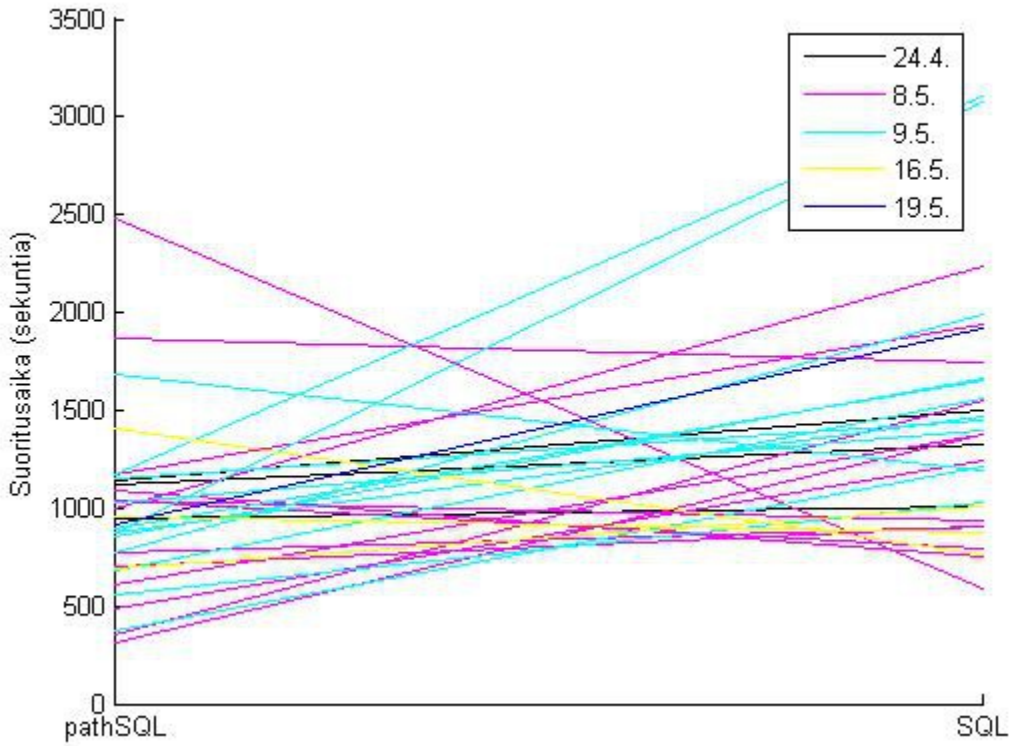
5.4. Suoritusajat

Suoritusaikojen kokonaistarkastelussa oli mukana 32 koehenkilöä. Kahdeksan koehenkilöä jätettiin pois vertailusta. Kaksi heistä oli tehnyt SQL-tehtävät kokonaan PathSQL-kielellä. Neljä koehenkilöä oli tehnyt osan SQL-tehtävistä PathSQL-kielellä, joten suoritusaikojen laskeminen heille oli ongelmallista. Lisäksi yksi koehenkilö oli yrittänyt ratkaista joitakin SQL-tehtäviä useammin kuin neljästi, ja yksi koehenkilö oli jättänyt PathSQL-sarjan kokonaan tekemättä.



Kuva 9. Erotukset PathSQL- ja SQL-tehtävien suoritusajoissa (SQL - PathSQL).

Laatikko-jana-kuvassa 9 on kuvattuna ero PathSQL-tehtävän ja sitä vastaavan SQL-tehtävän suoritusajassa (eli SQL-tehtävän suoritusajasta on vähennetty PathSQL-tehtävän suoritusaika) kaikkien 32:n koehenkilön osalta. Kuvassa on kuvattu pienin ja suurin arvo, ala- ja yläneljännekset sekä mediaani. Merkittävästi poikkeavat arvot on kuvattu erillisinä ympyröinä tai tähtinä, joissa koehenkilö on yksilöity numerolla. Kuvan perusteella vaikuttaa siltä, että koehenkilöt suoriutuivat PathSQL-tehtävistä hieman nopeammin kuin SQL-tehtävistä. Eroa suoritusajoissa testattiin Wilcoxonin merkittyjen sijalukujen testillä, josta saatiin p-arvoksi 0,001. Ero on tilastollisesti merkitsevä, eli koehenkilöt todella käyttivät PathSQL-tehtävien suorittamiseen vähemmän aikaa kuin SQL-tehtävien suorittamiseen.



Kuva 10. Koehenkilöiden PathSQL- ja SQL-tehtäväsarjojen kokonaissuoritusajat koetilaisuuden järjestämispäivän mukaan luokiteltuna (n=32).

Rinnakkaiskoordinaattikuvassa 10 nähdään jokaisen 32:n koehenkilön kaikkien PathSQL- ja SQL-tehtävien yhteenlasketut suoritusajat. Koehenkilöt on merkitty kuvassa eri väreillä sen mukaan, minä päivänä he osallistuivat kokeeseen. Päivän 28.4.2014 koetilaisuuden kaikkien koehenkilöiden suoritukset kuuluivat niihin aiemmin mainittuihin suorituksiin, jotka piti jättää pois

aikojen kokonaisvertailusta, joten he eivät ole myöskään mukana kuvassa. Viivan nouseva suunta kuvaa tilannetta, jossa koehenkilö on käyttänyt enemmän aikaa SQL- kuin PathSQL-tehtäväsarjan suorittamiseen ja viivan laskeva suunta kuvaa päinvastaista tilannetta. Kuvasta voidaan havaita, että useimmat koehenkilöt käyttivät PathSQL-tehtäviin vähemmän tai suunnilleen saman verran aikaa kuin SQL-tehtäviin. Yksi koehenkilö poikkeaa selkeästi tästä: hän käytti PathSQL-tehtäväsarjan suorittamiseen aikaa 2486 sekuntia eli hieman yli 40 minuuttia mutta SQL-tehtäviin ainoastaan 589 sekuntia eli noin kymmenen minuuttia.

Taulukko 17. Oikein ratkaistujen PathSQL- ja SQL-tehtävien suoritusaikojen vähimmäis- ja enimmäisarvot sekä keskiarvo ja mediaani (suoritusajat sekunteina).

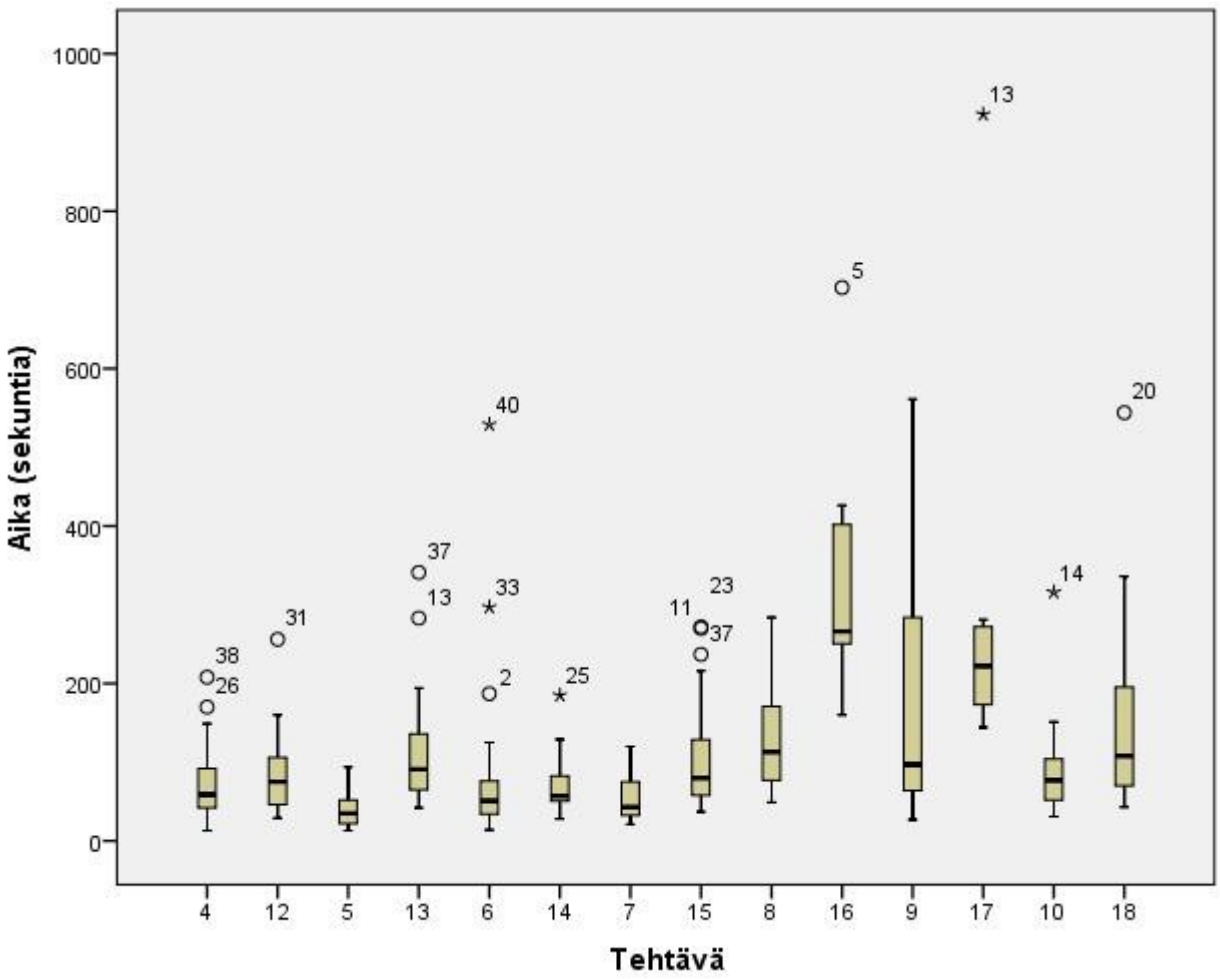
Tehtävä	n	Vähintään	Enintään	Keskiarvo	Mediaani	Tehtävä	n	Vähintään	Enintään	Keskiarvo	Mediaani
4	33	13	208	69,03	59	12	30	29	256	85,63	75
5	33	13	94	38,79	35	13	31	42	341	112,42	91
6	34	14	529	76,76	51	14	31	28	185	68,13	57
7	32	21	120	54,34	43	15	32	37	272	103,72	80
8	10	49	284	135,4	113	16	7	160	703	347,71	266
9	9	27	561	177,56	97	17	8	144	923	300,13	222
10	33	31	316	88,09	77	18	27	43	544	150,3	108

Taulukko 18. Väärin ratkaistujen PathSQL- ja SQL-tehtävien suoritusaikojen vähimmäis- ja enimmäisarvot sekä keskiarvo ja mediaani (suoritusajat sekunteina).

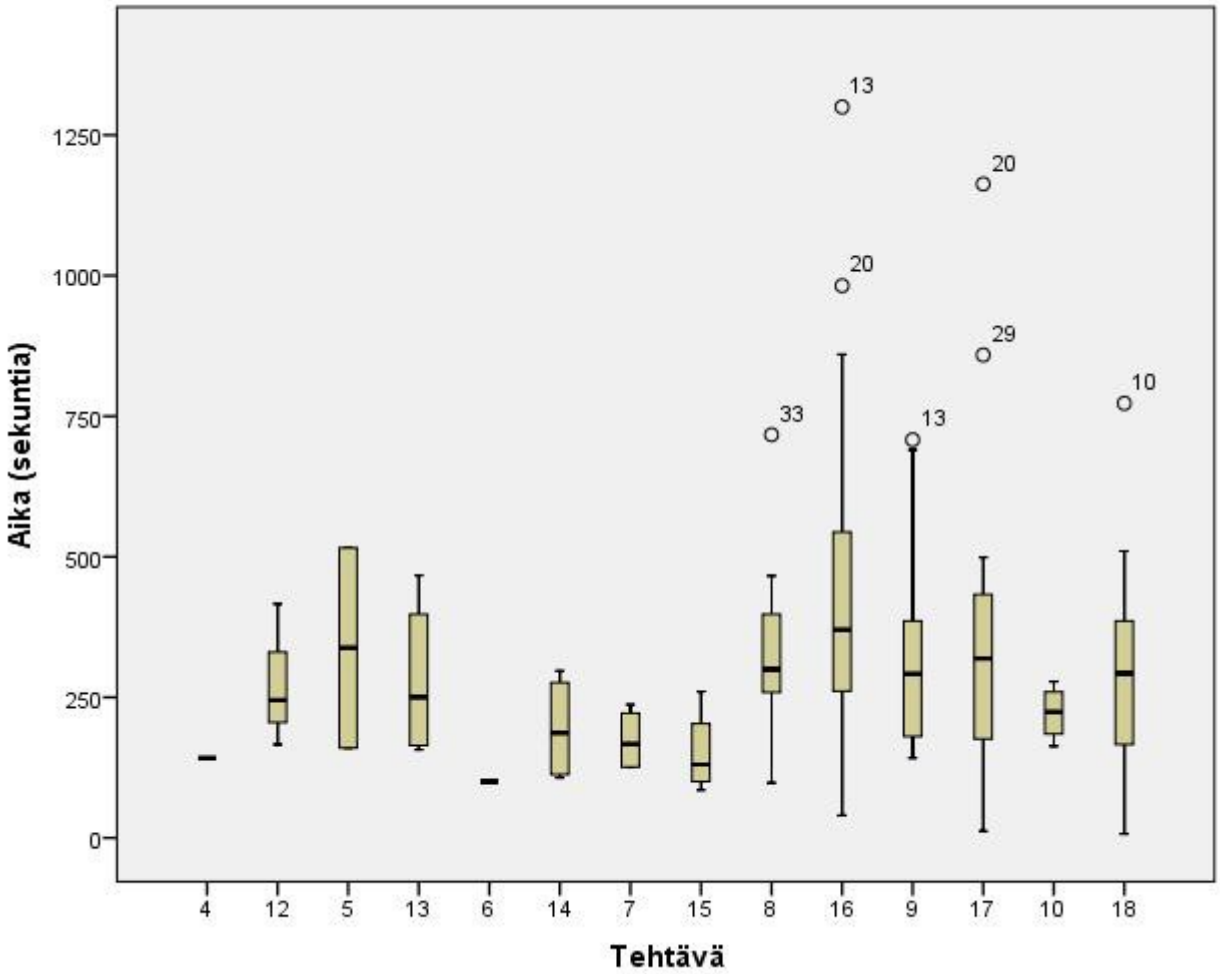
Tehtävä	n	Vähintään	Enintään	Keskiarvo	Mediaani	Tehtävä	n	Vähintään	Enintään	Keskiarvo	Mediaani
4	1	142	142	142	142	12	4	166	416	268	245
5	2	160	516	338	338	13	4	157	467	281,25	250,5
6	1	100	100	100	100	14	4	107	297	194,25	186,5
7	4	125	237	174	167	15	4	85	260	151,5	130,5
8	27	98	717	324,85	300	16	30	40	1300	439,43	370
9	28	142	708	328,54	292	17	29	12	1163	334,1	319
10	4	163	278	222,25	224	18	10	7	773	303,6	292,5

Eroa suoritusaikojen välillä vertailtiin myös tehtävittäin (PathSQL-tehtävä ja sitä vastaava SQL-tehtävä). Vertailussa mukana olleiden koehenkilöiden lukumäärä vaihteli eri tehtäväparien kohdalla (välillä 34-37). Taulukoissa 17 ja 18 on koehenkilöiden PathSQL- (tehtävät 4-10) ja SQL-tehtävien (tehtävät 12-18) oikeiden ja väärien vastausten tekoon käytetyn ajan pienimmät ja suurimmat arvot sekä keskiarvot ja mediaanit. Suoritusaikaan on laskettu kaikkiin ratkaisuyrityksiin yhteensä käytetty aika. Oikealla vastauksella tarkoitetaan sitä, että koehenkilö onnistui ratkaisemaan tehtävän oikein viimeistään viimeisellä (eli neljännellä) ratkaisuyrityskerralla. Muussa tapauksessa vastaus

on väärä. Taulukoista voidaan havaita, että koehenkilöt suoriutuivat PathSQL-tehtävistä nopeammin erityisesti silloin, kun tehtävän vastaus oli oikea.



Kuva 11. Oikein ratkaistujen tehtävien suoritusajat tehtävittäin.



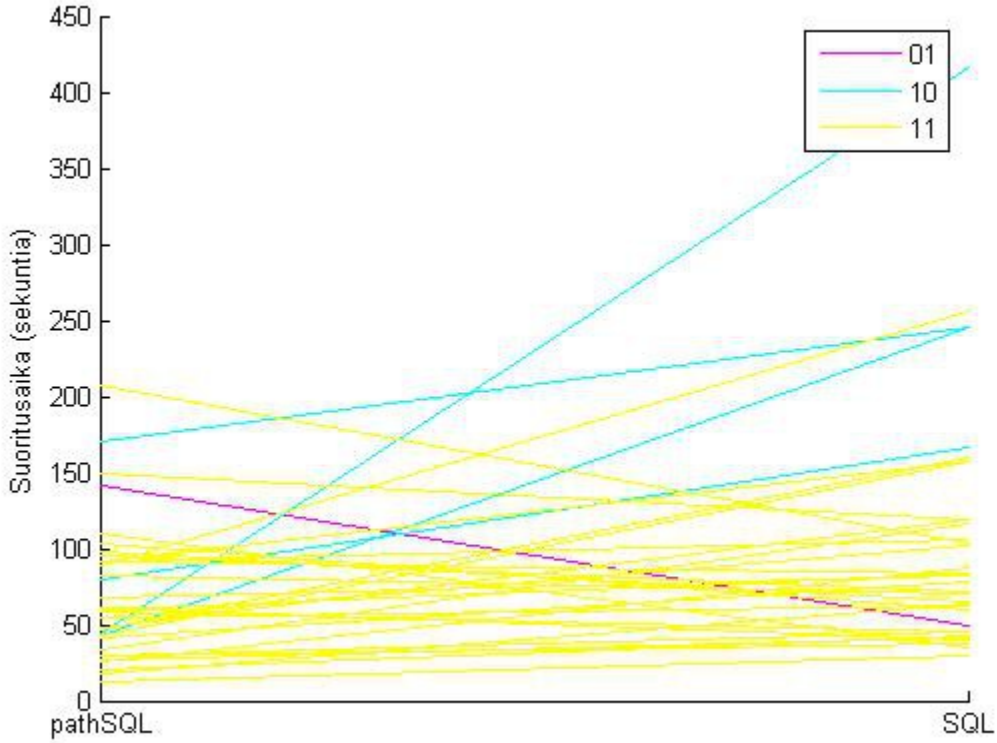
Kuva 12. Väärin ratkaistujen tehtävien suoritusajat tehtävittäin.

Laatikko-jana-kuvassa 11 on kuvattuna sellaisten koehenkilöiden suoritusajat, jotka suorittivat tehtävät oikein viimeistään neljännellä eli viimeisellä ratkaisuyrityskerralla. Kuvassa 12 on puolestaan kuvattuna sellaisten koehenkilöiden suoritusajat, jotka eivät onnistuneet ratkaisemaan tehtäviä oikein edes neljännellä ratkaisuyrityskerralla. Molemmissa kuvissa toisiaan vastaavat PathSQL- ja SQL-tehtävät ovat kuvattuina vierekkäin (esimerkiksi tehtävät 4 ja 12). Kuvissa on kuvattu tehtävien suoritusaikojen pienin ja suurin arvo, ala- ja yläneljännekset sekä mediaani. Merkittävästi poikkeavat arvot on kuvattu erillisinä ympyröinä tai tähtinä, joissa koehenkilö on yksilöity numerolla. Erityisesti onnistuneesti suoritettujen tehtävien suoritusajat vaikuttavat olevan kuvien perusteella lyhyempiä PathSQL-tehtävissä kuin SQL-tehtävissä. Epäonnistuneiden tehtävien suoritusajoissa ero ei ole yhtä selkeä.

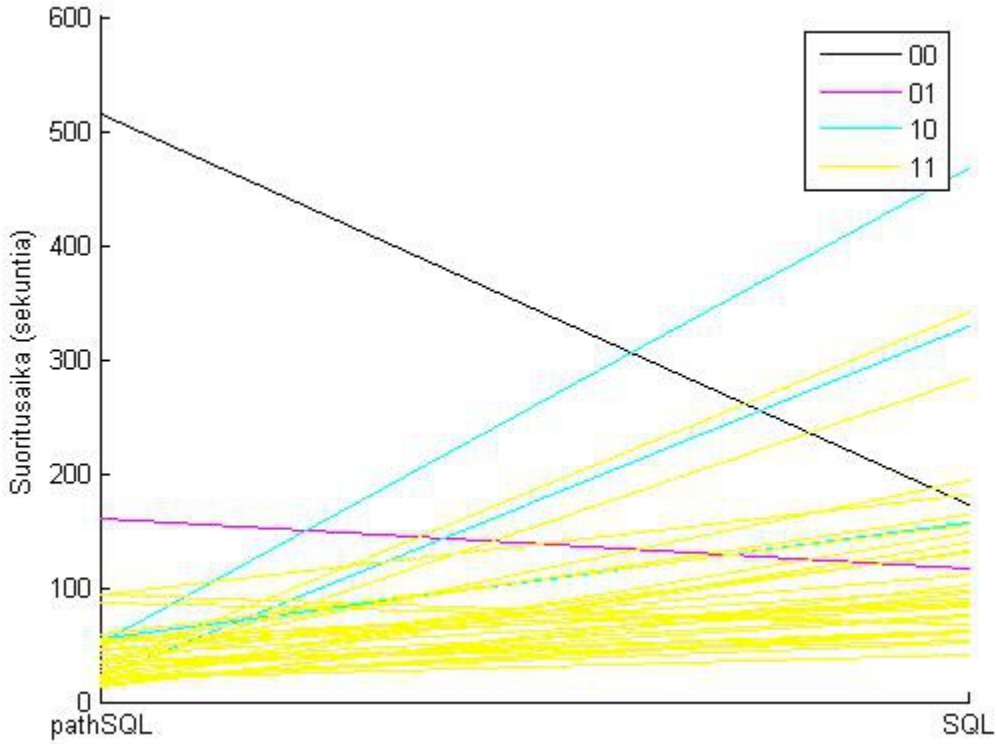
Taulukkoon 19 on koottu suoritusaikojen vertailu tehtäväpareittain. Kaikissa vertailuissa PathSQL-tehtävän suoritusaika osoittautui lyhyemmäksi kuin SQL-tehtävän. Eroja myös testattiin tilastollisesti Wilcoxonin merkittyjen sijalukujen testillä. Neljässä tapauksessa seitsemästä ero on tilastollisesti merkitsevä, kahdessa näistä jopa erittäin merkitsevä (tehtävät 5 ja 13 sekä tehtävät 7 ja 15). Lisäksi tehtävien 4 ja 12 välinen ero on melkein tilastollisesti merkitsevä. Polkuilmausten käyttö vaikuttaa siis nopeuttavan kyselyiden tekoa monimutkaisista hierarkioista.

Taulukko 19. Suoritusaikojen Wilcoxonin testien p-arvot tehtäväpareittain. Merkintä * tarkoittaa että p-arvo on tilastollisesti merkitsevä ja merkintä ** että arvo on tilastollisesti erittäin merkitsevä.

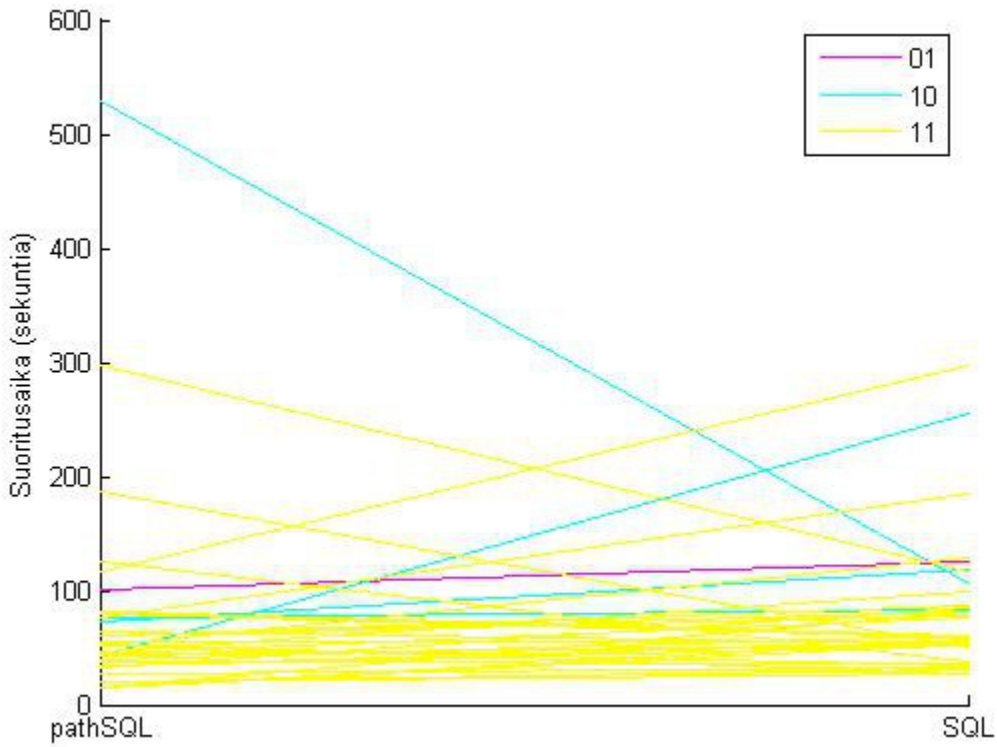
Tehtävät	**Koehenkilöiden lukumäärä**	**Pienempi suoritusaika**	**Wilcoxonin testin p-arvo**
4 ja 12	34	PathSQL	0,025
5 ja 13	35	PathSQL	0,000**
6 ja 14	35	PathSQL	0,061
7 ja 15	36	PathSQL	0,000**
8 ja 16	37	PathSQL	0,006*
9 ja 17	37	PathSQL	0,236
10 ja 18	37	PathSQL	0,009*



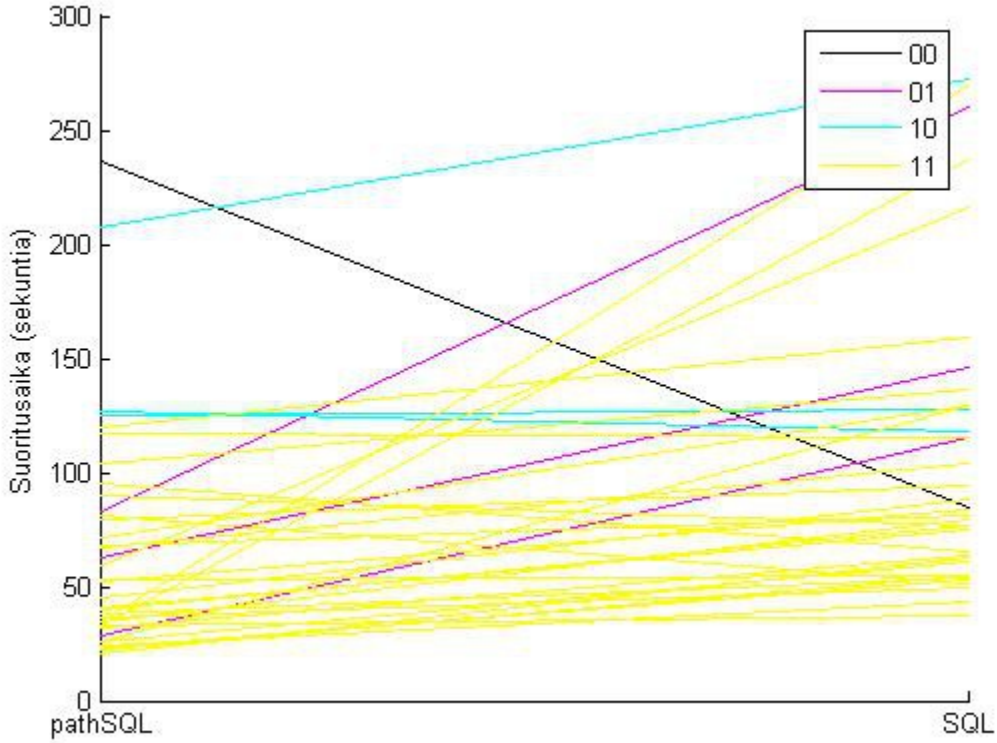
Kuva 13. Koehenkilöiden suoritusajat tehtäväparissa 4 ja 12 (n=34).



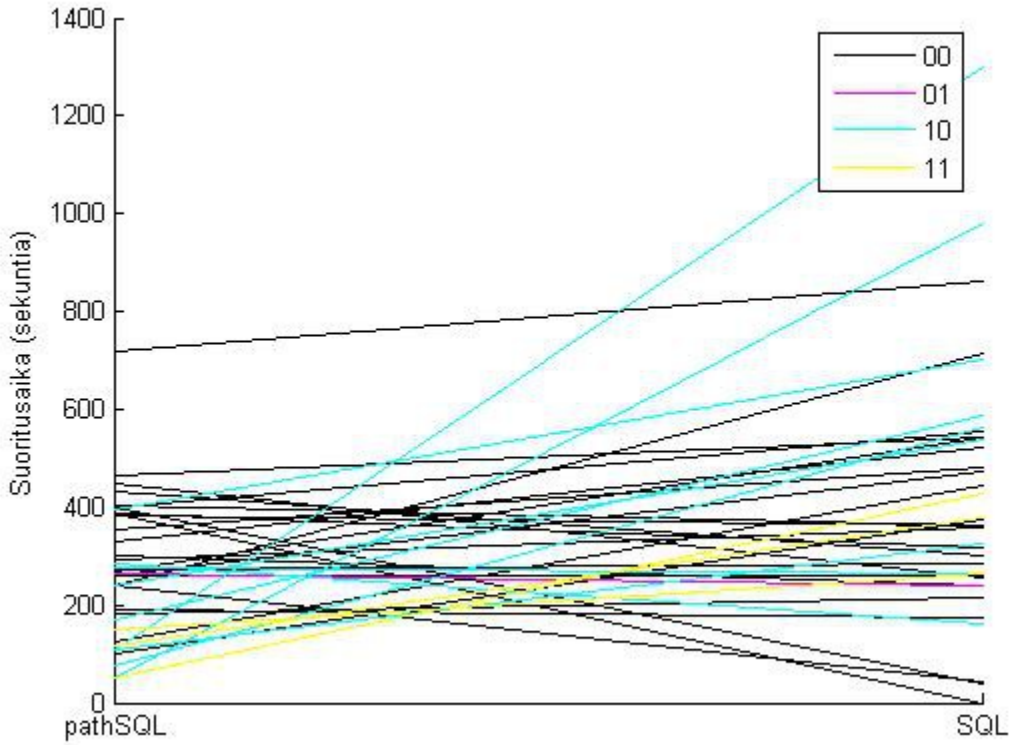
Kuva 14. Koehenkilöiden suoritusajat tehtäväparissa 5 ja 13 (n=35).



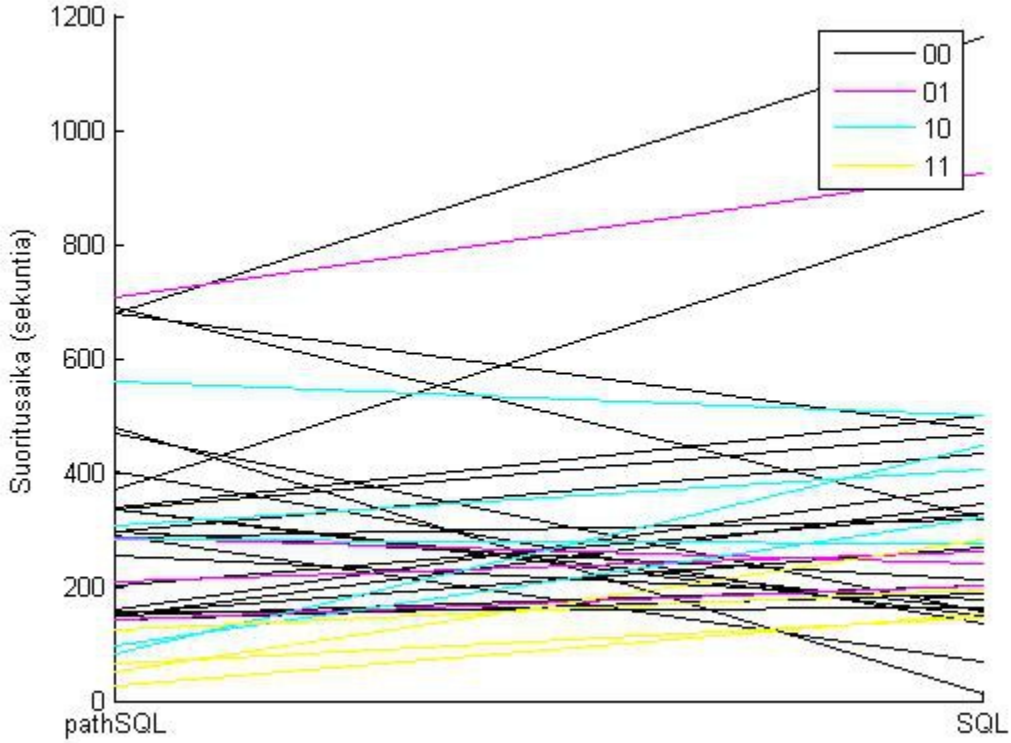
Kuva 15. Koehenkilöiden suoritusajat tehtäväparissa 6 ja 14 (n=35).



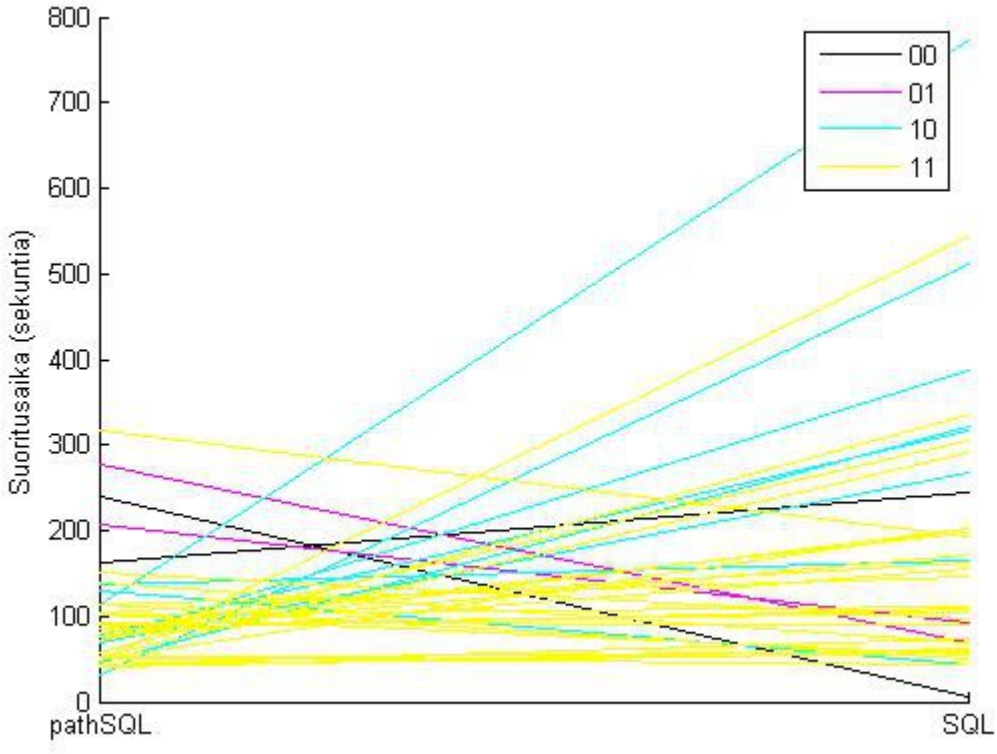
Kuva 16. Koehenkilöiden suoritusajat tehtäväparissa 7 ja 15 (n=36).



Kuva 17. Koehenkilöiden suoritusajat tehtäväparissa 8 ja 16 (n=37).



Kuva 18. Koehenkilöiden suoritusajat tehtäväparissa 9 ja 17 (n=37).



Kuva 19. Koehenkilöiden suoritusajat tehtäväparissa 10 ja 18 (n=37).

Rinnakkaiskoordinaattikuvissa 13-19 on koehenkilöiden toisiaan vastaavien PathSQL- ja SQL-tehtävien suoritusajat. Koehenkilöt on merkitty eri väreillä sen mukaan, onnistuivatko he molemmissa, vain SQL-tehtävässä, vain PathSQL-tehtävässä vai ei kummassakaan tehtävässä. Merkintä 00 tarkoittaa kohenkilöitä, jotka epäonnistuvat molemmissa tehtävissä, 01 koehenkilöitä, jotka onnistuivat ainoastaan SQL-tehtävässä, 10 koehenkilöitä, jotka onnistuivat ainoastaan PathSQL-tehtävässä ja 11 koehenkilöitä, jotka onnistuivat molemmissa tehtävissä. Viivan nouseva suunta tarkoittaa, että koehenkilö käytti enemmän aikaa SQL-tehtävän suoritukseen ja viivan laskeva suunta tarkoittaa, että koehenkilö käytti enemmän aikaa PathSQL-tehtävän suoritukseen. Kuvista voidaan nähdä, että useimmat koehenkilöt ovat käyttäneet SQL-tehtäviin enemmän tai saman verran aikaa kuin PathSQL-tehtäviin erityisesti niissä tapauksissa kun PathSQL-tehtävä on onnistunut ja SQL-tehtävä epäonnistunut. Toisin päin ilmiö ei ole yhtä voimakas: esimerkiksi kuvassa 16 vain SQL-tehtävässä onnistuneet koehenkilöt ovat käyttäneet SQL-tehtävän tekemiseen enemmän aikaa. Myös niissä tapauksissa, joissa molemmat tehtävät ovat onnistuneet, useimmat koehenkilöt ovat käyttäneet SQL-tehtäviin enemmän tai saman verran aikaa kuin PathSQL-tehtäviin. Kuvista voidaan havaita myös että niissä tapauksissa, joissa molemmat tehtävät ovat

onnistuneet, on useimmiten käytetty vähemmän aikaa kuin niissä tapauksissa, joissa molemmat tehtävät ovat epäonnistuneet.

6. Virheet PathSQL-tehtävien vastauksissa

Reisner [1981] mainitsee virheiden luokittelun ja koehenkilöille vaikeiden funktioiden tunnistamisen yhtenä keinona tukea kyselykielen suunnittelua. Virheanalyyseja tutkimuksissaan ovat käyttäneet esimerkiksi jo aiemmin luvussa 3 mainitut Reisner et al. [1975] sekä Welty ja Stemple [1981]. Molemmissa näistä tutkimuksista virheitä luokiteltiin erilaisiin luokkiin vakavuuden perusteella. Molemmissa myös syntaksivirheet, jotka olivat kielen syntaksin vastaisia, eroteltiin semanttisista virheistä, jotka olivat syntaksin mukaisia mutta tuottivat väärän tuloksen. Reisner et al. [1975] käyttävät syntaksivirheistä nimitystä muotovirhe ja Welty ja Stemple [1981] nimitystä kielivirhe. Molemmissa tutkimuksissa semanttisia virheitä kutsutaan sisältövirheiksi. Semanttisten virheiden luokittelua ovat käsitelleet esimerkiksi Brass ja Goldberg [2005]. He esittelevät tutkimuksessaan 43 eri tyyppistä semanttista SQL-virhettä.

Myös tässä tutkielmassa luokitellaan koehenkilöiden tekemiä virheitä, sillä näin saadaan arvokasta tietoa kielen helppokäyttöisyydestä ja opittavuudesta esimerkiksi jatkokehitystä tai opetusta varten. Koska tutkimuksen kohteena on PathSQL, keskitytään ainoastaan PathSQL-virheisiin.

6.1. Virheiden luokittelu

Koehenkilöiden PathSQL-kielen tehtäväsarjassa tekemät virheet luokiteltiin syntaksivirheisiin ja semanttisiin virheisiin. Kuten jo aiemmin mainittiin, syntaksivirheessä koehenkilön kysely ei ole PathSQL-kielen kieliopin mukainen. Semanttisessa virheessä kysely voi olla muodoltaan oikeaa PathSQL-kieltä, mutta tuottaa väärän tuloksen. Syntaksivirheet ja semanttiset virheet jaoteltiin eri alatyyppeihin, jotka on kuvattu tarkemmin taulukoissa 20 ja 21 sekä seuraavissa alaluvuissa 6.2 ja 6.3. Toisin kuin aiemmin mainituissa Reisnerin et al. [1975] sekä Weltyn ja Stemplen [1981] tutkimuksissa, virheitä ei määritelty vakavuusasteiltaan erilaisiksi vaan ainoastaan eri tyyppisiksi. Taulukoissa 20 ja 21 on listattu eri tyyppisten virheiden esiintymiskertojen lukumäärät koehenkilöiden ratkaisuyrityksissä sekä virheen ainakin kerran tehneiden koehenkilöiden lukumäärät. Yksi ratkaisuyritys saattoi sisältää useita eri virhetyyppejä. Jokaisesta virhetyypistä on laskettu ainoastaan yksi esiintyminen ratkaisuyritystä kohti: esimerkiksi kaksi semanttista virhettä liitosehdoissa on laskettu yhdeksi tyypin e liitosehtovirheeksi. Kaikki koehenkilöiden ratkaisuyrityskerrat on otettu mukaan tarkasteluun. Mukana ovat myös ne koehenkilöt ja ratkaisuyritykset, jotka on jostain syystä jätetty pois muista tilastollisista vertailuista. Yhteensä PathSQL-tehtävien ratkaisuyrityskertoja oli 549, joista 346 oli virheellisiä (sisälsi vähintään yhden syntaksivirheen tai semanttisen virheen tai oli kokonaan tyhjä). Virheellisten ratkaisuyritysten osuus oli siis noin 63 prosenttia eli yli puolet kaikista ratkaisuyrityksistä.

Taulukko 20. Eri tyyppisten syntaksivirheiden lukumäärät ja esiintymisprosentit ratkaisuyrityksissä sekä virheitä tehneiden koehenkilöiden lukumäärät.

Virhetyyppi	**Lukumäärä koehenkilöiden ratkaisuyrityksissä**	**Esiintymisprosentti kaikissa PathSQL-tehtävien ratkaisuyrityksissä**	**Virheen ainakin kerran tehneiden koehenkilöiden lukumäärä**
a) Kirjoitusvirhe relaatioiden tai terminaalien nimissä	27	4,9	10
b) Terminaali puuttuu	10	1,8	4
c) Väärin kirjoitettu ehto-osa	18	3,3	7
d) Virhe polun muodostamisessa			
i) Käytetään sulkunotaatiota yksittäiseen haaraan	16	2,9	8
ii) Muu virhe polun muodostamisessa	40	7,3	16
Yhteensä	111	20,2	

Taulukko 21. Eri tyyppisten semanttisten virheiden lukumäärät ja esiintymisprosentit ratkaisuyrityksissä sekä virheitä tehneiden koehenkilöiden lukumäärät.

Virhetyyppi	**Lukumäärä koehenkilöiden ratkaisuyrityksissä**	**Esiintymisprosentti kaikissa PathSQL-tehtävien ratkaisuyrityksissä**	**Virheen tehneiden koehenkilöiden lukumäärä**
a) Käytetty karteesista tuloa			
i) Ei käytetty polkuja ollenkaan	1	0,2	1
ii) Karteesinen tulo polkujen välillä	34	6,2	7
b) Annetaan polku väärään suuntaan	67	12,2	16
c) Käytetään sulkunotaatiota vaikka ei pitäisi	144	26,2	33
d) Käytetään merkintää / vaikka pitäisi käyttää merkintää //, tai välirelaatio puuttuu (esimerkiksi (institute/project))	38	6,9	17
e) Virheellinen liitosehto	43	7,8	14
f) Virhe polun muodostuksessa			
i) Polkua ei muodostettu kohteeseen asti	73	13,3	20
ii) Polku muodostettu liian ylhäältä	30	5,5	12
iii) Polkua ei muodostettu ollenkaan	3	0,5	2
g) Polkua ei haaroitettu	19	3,5	7
Yhteensä	452		

Syntaksivirheitä esiintyi yhteensä 111 ja semanttisia virheitä 452, eli semanttiset virheet olivat huomattavasti yleisempiä kuin syntaksivirheet. 310:ssa ratkaisuyrityksessä (56,5 prosenttia) esiintyi vähintään yksi semanttinen virhe, kun taas syntaksivirheitä sisältyi ainoastaan 104:ään ratkaisuyritykseen (18,9 prosenttia). Koehenkilöt osasivat siis melko hyvin tuottaa kieliopillisesti oikeita PathSQL-kyselyitä, mutta kielen semantiikka tuotti heille vaikeuksia. Erityisen hankalaa vaikutti olevan polun haaroittamiseen käytettävien sulkumerkkien oikeaoppinen käyttö: lähes kolmasosassa ratkaisuyrityksistä niitä käytettiin tarpeettomasti. Virheen teki ainakin kerran jopa 33

koehenkilöä eli enemmistö (82,5%) kaikista koehenkilöistä. Myös polun muodostaminen kohteeseen asti tuotti vaikeuksia: puolet koehenkilöistä teki tämän virheen ainakin kerran.

Koehenkilöiden PathSQL-tehtävien ratkaisuyrityksistä laskettiin myös tarpeettomat liitosehdot, jotka eivät aiheuta virhettä mutta joiden käyttö on kyselyssä turhaa. Näitä löydettiin yhteensä 176, joista 147 oli virheellisissä ratkaisuyrityksissä ja 29 oikein menneissä ratkaisuyrityksissä. Ratkaisuyritysten joukossa oli myös kaksi kokonaan tyhjää syötettä: toinen tehtävässä 4 ja toinen tehtävässä 8. Molemmat näistä olivat saman koehenkilön tekemiä.

6.2. Syntaksivirheet

Syntaksivirheillä on neljä alaluokkaa, jotka on merkitty kirjaimilla a, b, c ja d. Näistä d-luokan virheillä on kaksi alatyyppiä, joista käytetään merkintöjä d i ja d ii.

Tyyppi a (kirjoitusvirhe)

Tyypin a syntaksivirhe on virhe kyselyn relaatioiden tai terminaalien nimissä. Esimerkiksi taulujen nimissä esiintyvät kirjoitusvirheet kuuluvat tähän virhetyyppiin. Kolme koehenkilöä kirjoitti institute-relaation tilalle sanan ”institution” ja yksi heistä teki tämän virheen kahdessa eri tehtävässä kaikissa ratkaisuyrityksissään. Sanan ”insitute” kirjoitti institute-relaation nimeksi myös kolme koehenkilöä. Muita kirjoitusvirheitä olivat ”institue”, “deparment”, “prjoect”, ”employe” ja ”employeee”. Kahdeksan kirjoitusvirhettä 27:sta oli yhden käyttäjän tekemiä.

Tyyppi b (puuttuva terminaali)

Tyypin b syntaksivirheessä jokin kyselyn avainsana puuttuu. Tällaisia puuttuvia sanoja olivat FROM ja WHERE. Virhettä esiintyi neljällä koehenkilöllä kolmessa tehtävässä yhteensä 10 kertaa. FROM puuttui joistakin ratkaisuyrityksistä kahdelta koehenkilöltä ja WHERE kahdelta koehenkilöltä.

Tyyppi c (ehto-osan virhe)

Tyypin c syntaksivirheessä virhe on kyselyn ehto-osassa. Esimerkiksi numeroiden ja merkkijonojen vertaaminen ehto-osassa oli seuraava virhe: vertailu”institute.id='uta'” on syntaksin vastainen, sillä id on tietotyypiltään kokonaisluku, eikä sitä voi verrata merkkijonoon (”uta”). Myös esimerkiksi merkkijonosta saattoi puuttua toinen heittomerkki. Tyypin c syntaksivirheitä esiintyi 18:ssa ratkaisuyrityksessä.

Tyyppi d (polkuvirhe)

Tyypin d virheissä kyselyn polkua ei ole muodostettu syntaksin mukaisesti. Polkuvirheet on jaettu virheelliseen sulkumerkkien käyttöön (alatyyppi i) ja muihin polkuvirheisiin (alatyyppi ii).

Alatyypin i virheessä sulkumerkkejä on käytetty syntaksin vastaisesti yhteen polun haaraan. Esimerkiksi tehtävän 4 kyselyssä ”select institute.name, department.name from institute (/department)” ei olisi pitänyt käyttää sulkumerkkejä, sillä polussa on vain yksi haara.

Alatyypin ii virhe on esimerkiksi tehtävän 4 vastauksessa ”select institute.name, department.name from (/institute, institute/department)”. FROM-sanan jälkeen puuttuu sen relaation nimi, josta eteenpäin polku haaroitetaan sulkunotaatiolla. Lisäksi polun toisen haaran edestä puuttuu vinoviiva. Muita tämän tyypin virheitä ovat esimerkiksi ilmaisut ”institute,/department” (pilkun paikka kieliopin vastainen), ”institute (/, //project)” (ensimmäisessä polun haarassa ei ole relaation nimeä), ”institute(admin, department, //project)” (polun kahden ensimmäisen haaran edestä puuttuu vinoviiva), ”institute/%/%” (kieliopin vastaiset merkinnät polussa), ”institute//” (polun loppuosa puuttuu), ”employee(/project/department/institute/, /admin/institute)” (polun ensimmäinen haara päättyy vinoviivaan), ”department/employee,/project” (jälkimmäisen vinoviivan tai pilkun paikka kieliopin vastainen), ”//department,//admin” (ensimmäinen relaatio puuttuu polusta) ja ”institute(/employee, /department, /department.project, /admin)” (pisteen käyttö polussa), sekä toisen sulkumerkinnän puuttuminen polusta.

Alatyypin i virheitä esiintyi 16:ssa ja alatyypin ii virheitä 40:ssa ratkaisuyrityksessä.

6.3. Semanttiset virheet

Semanttisilla virheillä on seitsemän eri tyyppiä, jotka on merkitty kirjaimilla a, b, c, d, e, f ja g. Näistä a-tyypin virheillä on kaksi alatyyppiä, joista käytetään merkintöjä a i ja a ii. Lisäksi f-tyypin virheillä on kolme erilaista alatyyppiä, joista käytetään merkintöjä f i, f ii ja f iii.

Tyyppi a (karteesinen tulo)

Tyypin a semanttisessa virheessä on käytetty virheellisesti karteesista tuloa, mikä tuottaa kyselyn tulokseen (joka voi muuten olla oikein tai väärin) ylimääräistä ja väärää tietoa. Tällä virhetyypillä on kaksi alatyyppiä, joista toisessa (i) ei ole käytetty polkuja. Tätä virhetyyppiä esiintyi ainoastaan yhdessä tehtävän 4 vastauksessa: kyselyn ”select institute.name, department.name from institute, department” tulokseksi tulee karteesinen tulo kaikkien tietokannan instituutioiden (institute) ja osastojen (department) välillä. Jokainen instituutio on omalla rivillään jokaisen osaston kanssa riippumatta siitä kuuluuko osasto instituutioon vai ei.

Alatyypissä ii karteesisessa tulossa on mukana vähintään yksi polku. Esimerkiksi tehtävän 4 vastauskysely ”select institute.name, department.name from institute, institute/department” tuottaa karteesisen tulon oikean vastauksen rivien ja kaikkien tietokannan instituutioiden välillä. Alatyypin ii virheitä esiintyi 34:ssa ratkaisuyrityksessä.

Tyyppi b (polun väärä suunta)

Tyypin b virheessä kyselyssä annetun polun suunta on väärä. Esimerkiksi tehtävän 6 vastauksessa ”select project.name, department.name from project/department” polku on väärin päin, sillä osasto (department) sijaitsee hierarkiassa ylempänä kuin projekti (project): osastoilla on projekteja eikä toisin päin. Tyypin b virheitä esiintyi 67:ssa ratkaisuyrityksessä.

Tyyppi c (sulut)

Tyypin c virheessä käytetään sulkumerkintää, eli polun haaroitusta, vaikka ei pitäisi. Esimerkiksi tehtävän 8 vastauksessa ”select employee.name from institute(//employee, //admin)” koehenkilö on virheellisesti haaroittanut polun, vaikka ainoastaan polun ensimmäinen haara tuottaisi oikean tuloksen. Tyypin c virhe löytyi 144:sta ratkaisuyrityksestä.

Tyyppi d (askelvirhe)

Tyypin d virheessä kyselyn polun keskeltä puuttuu jokin relaatio, tai koehenkilö on käyttänyt yhtä vinoviivaa kahden sijaan. Esimerkiksi tehtävän 5 vastauksesta ”select institute.name, project.name from institute/project” puuttuu polun keskeltä relaatioiden institute ja project välistä relaatio department. Tehtävän 9 vastauksessa ”select employee.name from department/employee where department.name = 'sis'” on puolestaan käytetty yhtä vinoviivaa kahden sijaan. Tyypin d virhe oli 38:ssa ratkaisuyrityksessä.

Tyyppi e (liitosehtovirhe)

Tyypin e virhe tarkoittaa sitä, että kyselyssä on liitosehto, joka on syntaktisesti oikein mutta tuottaa virheellisen tuloksen. Esimerkiksi tehtävän 7 vastauskysely ”select project.name from institute//project where project.name='uta'” ei tuota oikeaa tulosta, sillä liitosehdossa määritellään “uta” projektin nimeksi, vaikka se pitäisi määritellä instituution nimeksi. 43:ssa ratkaisuyrityksessä oli tyypin e virhe.

Tyyppi f (polkuvirhe)

Tyypin f virheessä kyselyn polku on muodostettu virheellisesti siten, että sitä ei ole muodostettu kohteeseen asti (alatyyppi i), se on muodostettu liian ylhäältä (alatyyppi ii), tai se on jätetty kokonaan muodostamatta (alatyyppi iii). Esimerkiksi tehtävän 9 vastauksessa ”select employee.name from institute//project where department.name='sis'” koehenkilö ei ole muodostanut polkua loppuun (relaatioon employee) asti. Tehtävän 9 vastauksessa ”select employee.name from institute//employee where department.name = 'sis'” polku on muodostettu liian yltäältä (institute-relaatiosta, vaikka polun tulisi alkaa vasta department-relaatiosta). Tehtävän 8 vastauksessa ”select employee.name from employee” polku employee-relaatioon puuttuu kokonaan. Tyypin f virheitä

esiintyi ainoastaan tehtävien 8, 9 ja 10 ratkaisuyrityksissä. Alatyypin i virheitä oli ratkaisuyrityksissä 73 kappaletta, alatyypin ii 30 kappaletta ja alatyypin iii kolme kappaletta.

Tyyppi g (haaroitusvirhe)

Tyypin g virheessä kyselyn polkua ei ole haaroitettu oikein (tai ollenkaan). Esimerkiksi tehtävän 10 vastauksessa ”select institute.name, department.name, admin.name from institute/admin” on mukana ainoastaan polun toinen haara. Oikeassa vastauksessa, SELECT institute.name, department.name, admin.name FROM institute(/department, /admin), pitäisi olla myös haara institute/department. Virhetyyppiä esiintyi 19:ssa tehtävän 10 ratkaisuyrityksessä. Muiden tehtävien vastauksissa virhettä ei esiintynyt, sillä niissä koehenkilöiden ei tarvinnut haaroittaa kyselyn polkua.

6.4. Virheiden esiintyminen tehtävittäin

Virheitä tarkasteltiin myös tehtäväkohtaisesti jotta nähtäisiin, minkä tyyppisiä virheitä eri tehtävien vastauksissa esiintyi, ja olivatko jotkin virhetyypit tyypillisiä joillekin tietyille tehtäville. Taulukkoon 22 on listattu eri tyyppisten syntaksivirheiden esiintymismäärät eri tehtävien ratkaisuyrityksissä. Vastaavasti taulukossa 23 on listattu semanttiset virheet ja niiden esiintymismäärät tehtävittäin.

Taulukoissa 24 ja 25 on listattu kunkin virheen tehneiden koehenkilöiden lukumäärät tehtävittäin. Taulukoista voidaan nähdä, että yksittäisen virhetyypin teki tehtävää ratkaistessa pääsääntöisesti melko pieni määrä koehenkilöitä. Poikkeuksena tästä on syntaksivirheiden joukossa virhetyyppi d ii, jonka teki tehtävässä 10 koehenkilöä. Tähän virhetyyppiin (muu polkuvirhe) tosin lukeutuu hyvin monenlaisia virheitä ja tehtävä 8 oli kokeen tulosten perusteella koehenkilöille haastava. Tehtävissä 8 ja 9 monet koehenkilöt ovat tehneet monenlaisia semanttisia virheitä (tyyppejä b, c, d, e, f). Kuten tehtävä 8, myös tehtävä 9 oli useille koehenkilöille haastava.

Taulukossa 26 on listattu virheellisten ratkaisuyritysten lukumäärät tehtävittäin. Tehtävissä, joissa virheellisiä ratkaisuyrityksiä oli enemmän, myös erilaisten virheiden määrä oli suurempi.

Taulukko 22. Eri tyyppisten syntaksivirheiden lukumäärät tehtävien ratkaisuyrityksissä.

Tehtävä	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Tyyppi a (kirjoitusvirhe)	3	2	4	7	7	0	4
Tyyppi b (puuttuva terminaali)	0	2	0	7	0	1	0
Tyyppi c (ehto-osan virhe)	0	0	0	4	0	13	1
Tyyppi d i (polkuvirhe)	3	1	4	2	1	5	0
Tyyppi d ii (polkuvirhe)	4	2	0	2	18	5	9
Yhteensä	10	7	8	22	26	24	14

Taulukko 23. Eri tyyppisten semanttisten virheiden lukumäärät tehtävien ratkaisuyrityksissä.

Tehtävä	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Tyyppi a i (karteesinen tulo)	1	0	0	0	0	0	0
Tyyppi a ii (karteesinen tulo)	3	1	0	1	9	11	9
Tyyppi b (polun väärä suunta)	0	4	11	4	25	22	1
Tyyppi c (sulut)	2	2	6	0	89	45	0
Tyyppi d (askelvirhe)	0	1	0	2	5	30	0
Tyyppi e (liitosehtovirhe)	0	4	2	7	6	24	0
Tyyppi f i (polkuvirhe)	0	0	0	0	34	33	6
Tyyppi f ii (polkuvirhe)	0	0	0	0	0	30	0
Tyyppi f iii (polkuvirhe)	0	0	0	0	3	0	0
Tyyppi g (haaroitusvirhe)	0	0	0	0	0	0	19
Yhteensä	6	12	19	14	171	195	35

Taulukko 24. Eri tyyppisiä syntaksivirheitä tehneiden koehenkilöiden lukumäärät tehtävien ratkaisuyrityksissä.

Tehtävä	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Tyyppi a (kirjoitusvirhe)	2	1	2	4	3	0	2
Tyyppi b (puuttuva terminaali)	0	1	0	2	0	1	0
Tyyppi c (ehto-osan virhe)	0	0	0	1	0	5	1
Tyyppi d i (polkuvirhe)	3	1	2	1	1	5	0
Tyyppi d ii (polkuvirhe)	2	1	0	2	10	3	6

Taulukko 25. Eri tyyppisiä semanttisia virheitä tehneiden koehenkilöiden lukumäärät tehtävien ratkaisuyrityksissä.

Tehtävä	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Tyyppi a i (karteesinen tulo)	1	0	0	0	0	0	0
Tyyppi a ii (karteesinen tulo)	2	1	0	1	3	6	3
Tyyppi b (polun väärä suunta)	0	1	6	3	11	7	1
Tyyppi c (sulut)	1	1	4	0	27	20	0
Tyyppi d (askelvirhe)	0	1	0	2	3	13	0
Tyyppi e (liitosehtovirhe)	0	1	1	4	3	8	0
Tyyppi f i (polkuvirhe)	0	0	0	0	14	13	3
Tyyppi f ii (polkuvirhe)	0	0	0	0	0	12	0
Tyyppi f iii (polkuvirhe)	0	0	0	0	2	0	0
Tyyppi g (haaroitusvirhe)	0	0	0	0	0	0	7

Taulukko 26. Kaikkien sekä virheellisten ratkaisuyritysten lukumäärät tehtävittäin.

Tehtävä	Ratkaisuyritysten lukumäärä	Virheellisten ratkaisuyritysten lukumäärä
4	52	13
5	46	9
6	58	21
7	61	26
8	132	121
9	141	133
10	59	24

Alla on tarkasteltu virheiden esiintymistä jokaisessa PathSQL-tehtävässä sekä annettu joitakin esimerkkejä tehtävien virheellisistä ratkaisuyrityksistä virheineen sekä mahdollisine tarpeettomine liitosehtoineen.

Tehtävä 4

Tehtävässä 4 koehenkilöiden piti muodostaa polku relaatiohierarkian juuresta seuraavaan relaatioon. Tehtävä oli hyvin yksinkertainen, ja sen ratkaisuyrityksissä virheitä oli hyvin vähän. Myös virheellisten ratkaisuyritysten määrä oli vähäinen. Syntaksivirheitä ja semanttisia virheitä esiintyi suunnilleen yhtä paljon. Syntaksivirheistä jokaista virhetyyppiä esiintyi muutama, lukuun ottamatta puuttuviin avainsanoihin (tyyppi b) sekä ehto-osaan liittyviä virheitä (tyyppi c), joita ei ollut vastauksissa ollenkaan. Semanttisista virheistä esiintyi ainoastaan muutamia karteesiseen tuloon (tyyppi a) ja sulkujen tarpeettomaan käyttöön (tyyppi c) liittyviä virheitä. Taulukkoon 27 on listattu joitakin esimerkkejä tehtävän virheellisistä ratkaisuyrityksistä virhetyyppeineen.

Taulukko 27. Esimerkkejä tehtävän 4 virheellisistä ratkaisuyrityksistä virheluokkineen, sekä yrityksissä käytetyt tarpeettomat (mutta ei välttämättä virheelliset) liitosehdot.

Ratkaisuyritys	**Syntaksivirheet**	**Semanttiset virheet**	**Tarpeeton liitosehto**
select institute.name, department.name from insitute/department where institute.id = department.i_id	a		x
select institute.name, department.name from institute, institute/department		a ii	
select institute.name, department.name from (/institute, institute/department)	d ii	c	
select institute.name, department.name from institute, department		a i	
select institute.name, department.name from institute,/department where institute.id=department.i_id	d ii	a ii	x
select institute.name, department.name from institute(/department)	d i		

Tehtävä 5

Tehtävässä 5 piti muodostaa polku, jossa on yksi välirelaatio. Tehtävän ratkaisuyritykset eivät myöskään sisältäneet juurikaan virheitä. Taulukosta 26 voidaan nähdä, että tehtävässä oli pienin määrä virheellisiä ratkaisuyrityksiä (9) eli tehtävä oli koehenkilöille helppo. Semanttiset virheet (12 kappaletta) olivat hieman yleisempiä kuin syntaksivirheet (7 kappaletta). Syntaksivirheitä oli muutama jokaista tyyppiä lukuun ottamatta ehto-osan virheitä (tyyppi c). Semanttisista virheistä mukana oli muutamia tyyppien b (väärään suuntaan annettu polku), c (tarpeeton polun haaroitus) ja e (virhe liitosehdoissa) virheitä. Karteesiseen tuloon liittyviä virheitä (tyyppi a), polun väliosaan (tyyppi d) ja liitosehtoihin (tyyppi e) liittyviä virheitä oli kaikkia yksi. Taulukkoon 28 on listattu esimerkkejä tehtävän virheellisistä ratkaisuyrityksistä virhetyyppeineen.

Taulukko 28. Esimerkkejä tehtävän 5 virheellisistä ratkaisuyrityksistä virheluokkineen, sekä yrityksissä käytetyt tarpeettomat (mutta ei välttämättä virheelliset) liitosehdot.

Ratkaisuyritys	**Syntaksivirheet**	**Semanttiset virheet**	**Tarpeeton liitosehto**
select institute.name, project.name from employee//prjoect institute.id = project.id	a, b	b, e	x
select institute.name, project.name from institute/project		d	
select institute.name, project.name from institute, institute//project		a ii	
select institute.name, project.name from (institute, institute//project)	d ii	c	
select institute.name, project.name from institute (/, //project)	d ii	c	
select institute.name, project.name from institute (, //project)	d i		
select institute.name, project.name from institute//employee//project where project.id = department.id		b, e	x

Tehtävä 6

Tehtävässä 6 piti muodostaa polku, joka ei alkanut juuresta vaan alempana hierarkiassa olevasta relaatiosta. Koehenkilöiden piti lisäksi osata muodostaa polku oikeaan suuntaan. Tehtävän 6 ratkaisuyrityksissä semanttiset virheet olivat yleisempiä kuin syntaksivirheet. Virheellisiä ratkaisuyrityksiä oli yhteensä 21 eli melko vähän. Syntaksivirheitä esiintyi ainoastaan 8 kappaletta, joista puolet oli kahden käyttäjän tekemiä kirjoitusvirheitä (tyyppi a): sana “institute” oli kirjoitettu kaikissa tapauksissa väärin. Puolet virheistä oli sulkumerkkien käyttöä yhteen polun haaraan (tyyppi d i). Semanttisia virheitä oli puolestaan 19 kappaletta, joista lähes kaikki olivat tyyppiä b (polun suunta väärä) tai c (tarpeeton polun haaroitus). Tyypin b virheitä oli 11 kappaletta ja tyypin c virheitä 6 kappaletta. Tämän lisäksi tyypin e virheitä (virheellinen liitosehto) oli 2 kappaletta. Taulukossa 29 on esimerkkejä tehtävän virheellisistä ratkaisuyrityksistä virhetyyppeineen.

Taulukko 29. Esimerkkejä tehtävän 6 virheellisistä ratkaisuyrityksistä virheluokkineen, sekä yrityksissä käytetyt tarpeettomat (mutta ei välttämättä virheelliset) liitosehdot.

Ratkaisuyritys	**Syntaksivirheet**	**Semanttiset virheet**	**Tarpeeton liitosehto**
select project.name, department.name from project//department where project.id = department.id		b, e	x
select project.name, department.name from insitute//project where department.id=project.d_id	a		x
select project.name, department.name from institute (/department/project, /department)		c	
select project.name, department.name from institute(//project//department)	d i	b	
select project.name, department.name from institute/department (/project)	d i		

Tehtävä 7

Tehtävässä 7 polkuun piti liittää valintaehto. Tehtävän ratkaisuyrityksissä esiintyi hieman enemmän syntaksivirheitä (22 kappaletta) kuin semanttisia virheitä (14 kappaletta). Virheellisiä ratkaisuyrityksiä oli yhteensä 26. Syntaksivirheistä yleisimpiä olivat kirjoitusvirheet (tyyppi a) ja puuttuvat avainsanat, joita molempia esiintyi melko paljon (7 kappaletta). Tyypin a virheissä relaation institute tai department nimi oli kirjoitettu väärin. Myös liitosehdoissa ja poluissa oli joitakin syntaksivirheitä. Semanttisista virheistä yleisin oli liitosehtovirhe (tyyppi e), jota esiintyi 7 kertaa. Myös muutamia muita virhetyyppejä esiintyi muutamissa ratkaisuyrityksissä. Taulukossa 30 on esimerkkejä virheellisistä ratkaisuyrityksistä virhetyyppeineen.

Taulukko 30. Esimerkkejä tehtävän 7 virheellisistä ratkaisuyrityksistä virheluokkineen, sekä yrityksissä käytetyt tarpeettomat (mutta ei välttämättä virheelliset) liitosehdot.

Ratkaisuyritys	**Syntaksivirheet**	**Semanttiset virheet**	**Tarpeeton liitosehto**
select project.name from project(/department/institute) where institution.name = 'uta'	a, di	b	
select project.name from institute/project where institute.id = department.i_id and department.id = project.d_id and institute.name = 'uta'		d	x
select project.name from institute/department,/project where (institute.id='uta' and institute.id=department.i_id) and department.id=project.d_id	c, d ii	a ii	x
select project.name from institute//project, where institute.name = 'uta'	d ii		
select project.name project//institute where institute.name = 'sis' and project.d_id = department.id and department.i_id = institute.id	b	b, e	x
select project.name from institute//project where project.name='uta'		e	

Tehtävä 8

Tehtävässä 8 piti muodostaa polkuilmaus, joka sisältää useampia polkuja. Tehtävän 8 ratkaisuyritykset sisälsivät kaikkien tehtävien ratkaisuyrityksistä eniten syntaksivirheitä (26 kappaletta), mutta semanttisia virheitä oli vielä huomattavasti enemmän (171 kappaletta). Virheellisiä ratkaisuyrityksiä oli toiseksi eniten kaikista tehtävistä: 121 kappaletta, mikä on huomattavasti enemmän kuin useimmissa muissa tehtävissä. Syntaksivirheistä yleisin oli polkuvirhe (tyyppi d ii), jota esiintyi ratkaisuyrityksissä yhteensä 18 kertaa. Tyypin d i virheitä esiintyi kuitenkin vain yksi. Myös kirjoitusvirheitä relaatioiden nimissä (tyyppi a) esiintyi jonkin verran (7 kappaletta).

Semanttisista virheistä ylivoimaisesti yleisin (89 esiintymiskertaa) oli tyyppi c, eli sulkumerkintää käytettiin tarpeettomaan polun haaroittamiseen. Toiseksi yleisin semanttinen virhe oli se, ettei polkua muodostettu kohteeseen asti (f i). Tätä virhettä esiintyi 34 kertaa. Myös polun väärä suunta hierarkiassa (tyyppi b) oli yleinen (25 esiintymiskertaa). Muita semanttisia virheitä

esiintyi huomattavasti vähemmän (jokaista tyyppiä alle kymmenen kertaa). Taulukossa 31 on esimerkkejä virheellisistä ratkaisuyrityksistä virhetyyppeineen.

Taulukko 31. Esimerkkejä tehtävän 8 virheellisistä ratkaisuyrityksistä virheluokkineen, sekä yrityksissä käytetyt tarpeettomat (mutta ei välttämättä virheelliset) liitosehdot.

Ratkaisuyritys	**Syntaksivirheet**	**Semanttiset virheet**	**Tarpeeton liitosehto**
select employee.name from institute(//project, /admin)		c, f i	
select employee.name from employee(//department,/prjoect,/institute,/admin) where employee.i_id = institute.id and employee.p_id = project.id and employee.a_id = admin.id and employee.i_id = institute.id	a	b, c	x
select employee.name from institute(/employee, /department, /department.project, /admin) where (employee.i_id = institute.id or employee.d_id = department.id or employee.p_id = project.id or employee.id = admin.id) and department.i_id = institute.id or (department.i_id = institute.id and project.d_id = department.id) or admin.i_id = institute.id	d ii	c	x
select employee.name from institute(/department, /employee/project, /admin) where employee.id = institute.id		b, c, e	x
select employee.name from employee/admin, employee//project, employee//institute		a ii, b	
select employee.name from institute//employee where employee.i_id = institute.id or employee.d_id = department.id or employee.p_id = project.id or employee.a_id = admin.id		e	x
select employee.name from institute/*/employee		d	
select employee.name from employee		f iii	
select employee.name from institute(,/admin)/department/employee/project where institute.id=department.i_id and department.id=employee.d_id and employee.i_id=institute.id and admin.i_id=institute.id and project.d_id=department.id	d i, d ii	b	x
select employee.name from institute//	d ii	f i	

Tehtävä 9

Tehtävässä 9 koehenkilöiden piti muodostaa useita polkuja sisältävä polkuilmaus ja liittää siihen valintaehto. Tehtävässä 9 virheellisiä ratkaisuyrityksiä oli eniten: 133 kappaletta. Syntaksivirheitä näissä esiintyi 24 kertaa, ja semanttisia virheitä oli enemmän kuin yhdenkään muun tehtävän ratkaisuyrityksissä (195 kappaletta). Syntaksivirheistä yleisin oli virhe liitosehdoissa (tyyppi c) jota esiintyi 13 kertaa. Tämän lisäksi erilaisia polkuvirheitä (tyyppi d) esiintyi yhteensä 10 kertaa, ja avainsana puuttui yhdestä ratkaisuyrityksestä (tyyppi b).

Kuten tehtävässä 8, myös tehtävän 9 ratkaisuyrityksissä yleisin semanttinen virhe oli tarpeeton polun haaroittaminen sulkumerkinnän avulla (tyyppi c). Myös tyyppejä b (polun väärä suunta), d (puutteellinen polun väliosa), e (väärä liitosehto) sekä tyypin f alatyyppejä i ja ii (polku muodostettu liian ylhäältä hierarkiasta tai jätetty muodostamatta kohderelaatioon asti) esiintyi melko runsaasti (noin 20-30 tyyppiä kohti). Lisäksi karteesiseen tuloon (tyyppi a ii) liittyviä virheitä esiintyi 11 kertaa. Taulukossa 32 on esimerkkejä virheellisistä ratkaisuyrityksistä virhetyyppeineen.

Taulukko 32. Esimerkkejä tehtävän 9 virheellisistä ratkaisuyrityksistä virheluokkineen, sekä yrityksissä käytetyt tarpeettomat (mutta ei välttämättä virheelliset) liitosehdot.

Ratkaisuyritys	**Syntaksivirheet**	**Semanttiset virheet**	**Tarpeeton liitosehto**
select employee.name from department/project where department.name = 'sis'		d, f i	
select employee.name from institute(/employee, /department/project) where department.name = 'sis'		c, f ii	
select employee.name from department//employee department.name = 'sis'	b		
select employee.name from institute//employee where employee.d_id='sis' or employee.p_id='sis'	c	e, f ii	x
select employee.name from institute(//department//project) where project.name = 'sis'	d i	e, f i	
select employee.name from institute(/employee, /*/*/employee) where institute.name='sis'		c, e, f ii	
select employee.name from department/employee,/project where department.id=employee.d_id and project.id=employee.p_id and project.d_id=department.id and department.name='sis'	d ii	a ii	x
select employee.name from employee//department//project where department.name = 'sis' and where employee.d_id = department.id or where employee.p_id = project.id and project.d_id = department.id and where department.name = 'sis'		b	x
select employee.name from institute/employee		d, f ii	
select employee.name from employee(/project,/department) where department.name='sis'		b, c	
select employee.name from institute (/employee, /department, //project) where department.name = 'sis'		c, f ii	
select employee.name from institute/department/employee where department.name = 'sis'		d	

Tehtävä 10

Tehtävässä 10 piti haaroittaa polku. Tehtävässä oli 24 virheellistä ratkaisuyritystä. Syntaksivirheitä niissä esiintyi 14 kertaa, joista 9 oli polkuvirheitä (tyyppi d ii). Lisäksi ratkaisuyrityksissä oli kirjoitusvirheitä, joista 4 esiintyi relaatioiden nimissä (tyyppi a) ja yksi liitosehdoissa (tyyppi c). Kaikissa tyypin a virheissä institute-relaation nimi oli kirjoitettu väärin.

Semanttisia virheitä oli yhteensä 35 kappaletta, ja niistä eniten esiintyi tyyppiä g, eli polku oli haaroitettu väärin tai haaroittamista ei tehty ollenkaan. Tätä virhetyyppiä ei esiintynyt ollenkaan muiden tehtävien ratkaisuyrityksissä, sillä tehtävä 10 oli ainoa tehtävä, jossa polun haaroittamista tarvittiin. Tyypin f i virheitä, joissa polkua ei muodostettu kohteeseen asti, oli yhteensä 6. Lisäksi tyyppien b (polun väärä suunta) esiintyi yksi virhe. Taulukossa 33 on joitakin esimerkkejä virheellisistä ratkaisuyrityksistä virhetyyppeineen.

Taulukko 33. Esimerkkejä tehtävän 10 virheellisistä ratkaisuyrityksistä virheluokkineen, sekä yrityksissä käytetyt tarpeettomat (mutta ei välttämättä virheelliset) liitosehdot.

Ratkaisuyritys	**Syntaksivirheet**	**Semanttiset virheet**	**Tarpeeton liitosehto**
select institute.name, department.name, admin.name from institute/department/admin) where institution.id = department.i_id and institution.id = admin.i_id	a, d ii	g	x
select institute.name, department.name, admin.name from institute/admin		f i, g	
select institute.name, department.name, admin.name from //department,//admin where institute.id=department.i_id and institute.id=admin.i_id	d ii	a ii, g	x
select institute.name, department.name, admin.name from institute, (/department, /admin)	d ii		
select institute.name, department.name, admin.name from institute/department/employee/admin		b, g	
select institute.name, department.name, admin.name from institute//	d ii	f i, g	
select institute.name, department.name, admin.name from institute(/department, /admin) where institute.id department.i_id and department.i_id = institute.id	c		x
select institute.name, department.name, admin.name from institute//admin where institute.id = department.i_id and department.i_id = admin.i_id		f i, g	x

7. Pohdinta

Luvussa tulkitaan aiemmissa luvuissa esiteltyjä kokeen tuloksia, pohditaan kokeeseen liittyviä ongelmia, sekä esitetään joitakin PathSQL-kielen jatkokehitykseen ja tuleviin käyttäjäkokeisiin liittyviä ehdotuksia.

7.1. Koeasetelman ongelmat

Koeasetelmassa oli useita asioita, jotka saattoivat vaikuttaa kokeen tulokseen. Koehenkilöt suorittivat kokeen melko vapaasti ja omaan tahtiinsa ilman erityistä valvontaa ja kontrollia, ja tämä johti joidenkin koehenkilöiden kohdalla esimerkiksi siihen, että osa tai kaikki tehtävistä tehtiin väärällä kielellä, tai että ratkaisuyrityksiä oli liian monta. Tämä olisi voitu estää esimerkiksi lisäämällä valvontaa koetilanteessa, tai korjaamalla järjestelmää siten, ettei ratkaisuyrityksiä voi olla missään tilanteessa sallittua määrää enemmän. Järjestelmää voisi kehittää myös sellaiseksi, että se antaisi käyttäjille enemmän palautetta, esimerkiksi virheellisen tuloksen tai virheilmoituksia. Järjestelmän antama palaute voisi vähentää virheitä, koska käyttäjät voisivat korjata kyselyitään palautteen perusteella. Näin voitaisiin välttyä ainakin pienimmiltä kirjoitusvirheiltä ja muilta huolimattomuudesta johtuvilta virheiltä.

Myös se, että koetilaisuutta edeltävää PathSQL-kieleen tutustumista ei kontrolloitu, saattoi vaikuttaa kokeen tulokseen. Kaikki koehenkilöt eivät osallistuneet kieltä käsittelevälle luennolle eivätkä välttämättä tutustuneet edes luentomateriaaliin kurssin kotisivulla. Osa saattoi puolestaan tutustua kieleen ennen koetta hyvinkin tarkasti. Ero koehenkilöiden motivaatiossa saattoi vaikuttaa kokeen tulokseen muutenkin: jotkut koehenkilöistä tekivät koetehtävät ehkä hyvinkin huolellisesti, kun osa taas saattoi suhtautua kokeeseen ”pakollisena pahana” ja tehdä tehtävät huolimattomasti ja mahdollisimman nopeasti. Toisaalta kokeen suoritusaikojen perusteella on pikemminkin niin päin, että tehtävän oikein suorittanut koehenkilö usein myös käytti sen tekemiseen vähemmän aikaa kuin tehtävässä epäonnistunut henkilö.

Koehenkilöiden määrä oli melko vähäinen, ja osa koesuorituksista jouduttiin jättämään pois kokeen tulosten tarkasteluista erilaisista syistä. Tämä heikentää kokeen tulosten yleistettävyyttä: tuloksista ei voida tehdä kovinkaan varmoja tai pitkälle vietyjä johtopäätöksiä kielen mahdollisten käyttäjien toiminnasta.

Kokeessa tutkittiin ainoastaan koehenkilöiden suoritusten oikeellisuutta ja suoritusnopeutta, eikä puututtu siihen, miten koehenkilöt uuden kyselykielenkielen kokivat: pitivätkö he PathSQL-kieltä esimerkiksi helpompana vai vaikeampana käyttää kuin tavallista SQL-kieltä, tai mitkä kielen ominaisuudet he kokivat hyödyllisiksi, hankaliksi tai kyselyiden tekoa helpottaviksi. Tämä olisi voitu toteuttaa esimerkiksi kyselylomakkeella kokeen suorituksen jälkeen.

7.2. Kokeen tuloksesta

Koehenkilöt ratkaisivat tehtäviä hieman enemmän oikein PathSQL-kielellä kuin SQL-kielellä, mutta ero ei ollut tilastollisesti merkitsevä. Sen sijaan PathSQL-kielellä tehtävien ratkaisuun kului vähemmän aikaa, ja ero oli erityisen selkeä oikein ratkaistujen tehtävien kohdalla. Syntaktisesti ja semanttisesti oikeanlaisten SQL-kielisten kyselyiden muotoilu monine liitosoperaatioineen ja liitosehtoineen vie siis enemmän aikaa kuin PathSQL-kielen lyhyiden polkuilmaisujen käyttö. Sen sijaan jos PathSQL-kyselyihin liitetään virheellisesti tai muuten tarpeettomasti esimerkiksi SQL-kyselyille ominaisia liitosehtoja, kyselyyn käytetty aika kasvaa (ja tulos on usein virheellinen). Vaikuttaa siis siltä, että PathSQL-kielen käyttö hierarkkisessa datassa säästää käyttäjien aikaa.

Kielten vertailu kokeessa ei ollut tasapuolista samalla tavalla kuin monissa luvussa 3 mainituissa varhaisissa kyselykielitutkimuksissa, joissa molemmat kokeessa tutkituista kielistä olivat koehenkilöille ennestään tuntemattomia. Tämän tutkimuksen koehenkilöt ovat opiskelleet SQL-kielen käyttöä vähintään yhdellä (useimmat ainakin kahdella) yliopiston kurssilla ja osa heistä on käyttänyt kieltä mahdollisesti työssä tai vapaa-ajallaan, kun taas PathSQL on heille täysin uusi kieli, jonka opetukseen ei käytetty juurikaan aikaa verrattuna esimerkiksi moniin luvussa 3 esitellyistä tutkimuksista, joissa kyselykielien opetukseen oli järjestetty useita opetustilaisuuksia harjoitustehtävineen. SQL-kielen osaaminen saattoi vaikuttaa kokeen tulokseen monella tavalla ja saattoi toisaalta parantaa PathSQL-tehtävien suoritusta mutta toisaalta myös heikentää sitä. PathSQL perustuu SQL-kieleen ja jakaa monia sen piirteitä, mutta toisaalta jotkin SQL-kielen piirteet poikkeavat PathSQL-kielestä, jolloin niiden soveltaminen PathSQL-kieleen aiheuttaa virheitä. Tällaisia ovat esimerkiksi WHERE-osan liitosehdot, joiden käyttö on usein välttämätöntä hierarkkisissa SQL-kyselyissä, mutta jotka voivat aiheuttaa virheitä PathSQL-kyselyissä. Kurssiarvosanojen ja koetuloksen vertailun sekä oikein ratkaistujen PathSQL- ja SQL-tehtävien lukumäärien vertailun perusteella vaikuttaa kuitenkin siltä, että hyvin SQL-kieltä osaavat koehenkilöt oppivat PathSQL-kielen käytön paremmin. Toisaalta hyvä kurssiarvosana voi kertoa myös henkilön korkeasta motivaatiosta opiskelua kohtaan, ja tällaiset henkilöt ehkä olivat motivoituneempia myös tässä kokeessa, joka oli osa kurssin suoritusta. Toisaalta hyvä kurssiarvosana ja nopea uuden kyselykielen oppiminen saattavat merkitä yksinkertaisesti vain sitä, että henkilö on nopea oppija tai tietoteknisesti taitava. Kolme koehenkilöistä ei onnistunut suorittamaan onnistuneesti juurikaan SQL-tehtäviä, mutta onnistui kuitenkin melko hyvin PathSQL-tehtäväsarjassa, kun taas toisin päin vastaavaa ei ollut juurikaan havaittavissa. Tämä saattaa viitata siihen, että PathSQL voisi sopia myös SQL-kieltä heikosti tai ei ollenkaan osaavien käyttäjien tarpeisiin: siis myös muille kuin tietotekniikan ammattilaisille. Kokeen toistaminen täysin ohjelmointitaidottomia ja kyselykielten käytössä kokemattomia koehenkilöitä käyttäen voisi

tuottaa kiinnostavia tuloksia, varsinkin jos PathSQL-kieltä kehitetään jatkossa noviisikäyttäjien tarpeisiin.

7.3. Huomioita virheistä

Koehenkilöt tekivät eniten virheitä PathSQL-kyselyissä, joissa polun ensimmäisen ja viimeisen relaation väliin tuli useampi kuin yksi erilainen polku. Tällaisia olivat tehtävät 8 (SELECT employee.name FROM institute//employee) ja 9 (SELECT employee.name FROM department//employee WHERE department.name = 'sis'), joissa molemmissa employee-relaatioon johtaa useampi kuin yksi polku. Tyypillisesti koehenkilöt yrittivät haaroittaa polun useaksi employee-relaatioon johtavaksi poluksi (tyypin c semanttinen virhe), vaikka //-operaattori olisi kattanut kaikki alku- ja loppurelaation väliset polut. Kahden vinoviivan käytön ongelmallisuus saattaa johtua siitä, että se on koehenkilöille ennestään tuntematon, kun taas yhtä vinoviivaa käytetään laajasti jopa arkipäiväisessä tiedonkäsittelyssä (esimerkiksi internet) samalla tavalla kuin PathSQL-kielessä. PathSQL-kielessä //-operaattoria käytetään samaan tapaan kuin XML-kyselykieli XPathissa, jota sivutaan aineopintoihin kuuluvalla Tietokantaohjelmointi-kurssilla mutta johon perehdytään tarkemmin vasta syventävien opintojen vaiheessa. Koehenkilöt osallistuivat kokeeseen osana Tietokantaohjelmointi-kurssia, joten on todennäköistä, ettei ainakaan enemmistöllä heistä ole juurikaan aiempaa kokemusta XPathin tai muiden XML-kyselykielten käytöstä. Toisaalta monet koehenkilöistä käyttivät //-operaattoria onnistuneesti helpommissa tehtävissä, kuten tehtävässä 5, jossa kyselyn polun saattoi kirjoittaa muotoon institute//project pitemmän muodon institute/department/project sijaan. Hankaluudet tehtävissä 8 ja 9 saattavat siis johtua jostain muusta kuin //-operaattorin käytön vaikeudesta: koehenkilöt mahdollisesti ymmärsivät tehtävänannot niin, että niissä vaaditaan polkujen haaroitusta. Kokeen uudelleen suorittaminen esimerkiksi XML-kyselykieliä käsittelevän kurssin opiskelijoita koehenkilöinä käyttäen voisi tarjota kiinnostavan vertailukohteen tämän tutkielman kokeen tuloksille: tällöin voitaisiin tutkia selviytyvätkö polkuorientoituneita kieliä aiemmin käyttäneet käyttäjät paremmin tehtävistä, jotka edellyttävät tehtävien 8 ja 9 tavoin useamman samaan relaatioon johtavan polun käyttöä kyselyssä.

Koehenkilöiden tehtävässä 8 tekemistä virheistä ainakin osa saattoi johtua tehtävänannon väärinymmärryksestä: koehenkilöt saattoivat käsittää tehtävänannon niin, että kyselyn tulisi tuottaa tulokseksi työntekijät, jotka ovat töissä kaikissa tehtävänannossa mainituissa paikoissa (instituutio, osasto, projekti ja hallintoelin). Tällöin koehenkilöt saattoivat esimerkiksi lisätä kyselyyn ylimääräisiä liitosehtoja, jotka aiheuttivat virheellisen tuloksen. Mahdollisissa tulevissa käyttäjäkokeissa on siis syytä kiinnittää huomiota siihen, että tehtävänannot ovat mahdollisimman selkeitä ja yksiselitteisiä.

Koehenkilöt kirjoittivat jonkin verran relaatioiden nimiä väärin. Tätä olisi voinut ehkä ainakin osittain ehkäistä sillä, että tietokannan ja kyselyiden kieli olisi ollut sama kuin tehtävänannoissa.

Nyt tietokannan relaatiot olivat englanninkielisiä, kun taas tehtävänantojen kieli oli suomi. Esimerkiksi tehtävänannossa puhuttiin instituutioista, kun taas tietokannassa vastaavan relaation nimi oli institute. Osa koehenkilöistä kirjoitti relaation nimen muotoon ”institution”, joka on sanan instituutio käytetympi käännös.

Liitosehtojen virheellinen tai tarpeeton käyttö oli myös melko yleistä. Esimerkiksi monet koehenkilöt lisäsivät ehtojen määrää, jos ensimmäiset ratkaisuyritykset eivät tuottaneet oikeaa vastausta. Kuten jo aiemmin on mainittu, tämä saattaa johtua siitä, että SQL on koehenkilöille ennestään tuttu mutta PathSQL ei, ja SQL-kielessä kyselyt monimutkaisista hierarkioista vaativat runsasta liitosehtojen käyttöä.

Koehenkilöt tekivät melko paljon polun suuntavirheitä, eli kirjoittivat polun relaatiot joko osittain tai kokonaan väärässä järjestyksessä. Esimerkiksi tehtävässä 6 polun suuntavirhe löytyi kuuden koehenkilön ratkaisuyrityksistä. He olivat sijoittaneet project-relaation polkuun ennen department-relaatiota, vaikka project-relaatio sijaitsee relaatioiden muodostamassa hierarkiassa department-relaation alapuolella. Tehtävää 6 vastaavassa SQL-tehtävässä 14 nämä kuusi koehenkilöä olivat kirjoittaneet jokaiseen ratkaisuyritykseensä project-relaation ennen department-relaatiota, esimerkiksi: "select project.name, department.name from project, department where project.d_id = department.id". Myös ratkaisuyrityksissä, joissa käytettiin JOIN-operaatioita project-relaatio oli kirjoitettu ensin. SQL-kielessä relaatioiden järjestyksellä ei ole tuloksen kannalta merkitystä, kun taas PathSQL-kielessä polun väärä suunta aiheuttaa semanttisen virheen, ja useimmiten kysely tuottaa tällöin tyhjän tuloksen. Tämä olisi ehkä hyvä ottaa huomioon PathSQL:n jatkokehityksessä, erityisesti jos kieltä kehitetään SQL-kieltä jo osaavien käyttäjien tarpeisiin.

8. Yhteenveto

SQL-relaatiotietokanta on ylivoimaisesti suosituin ja käytetyin tietokantatyyppi. Monella sovellusalueella tarvitaan hierarkkiseen muotoon järjestettyä tietoa, ja relaatiotietokannoissa tämä tieto tallennetaan relaatioihin, jotka muodostavat hierarkkisen rakenteen. Kyselyt tällaisesta tietokannasta voivat kuitenkin olla SQL-kyselykielellä pitkiä ja monimutkaisia ja vaatia useiden liitosehtojen, operaatioiden ja joskus jopa alikyselyiden käyttöä. Käyttäjän tulee myös tuntea tarkasti tietokannan rakenne pystyäkseen navigoimaan tehokkaasti hierarkian eri tasojen välillä. SQL-kieleen perustuva PathSQL-kyselykieli pyrkii helpottamaan ja nopeuttamaan kyselyjen tekoa hierarkkisessa muodossa olevasta datasta korvaamalla pääavain-vierasavainparien ja liitosoperaatioiden käytön polkuilmaisujen käytöllä. Erityisesti PathSQL-kielen //- ja *-operaattorien käyttö mahdollistaa sen, ettei käyttäjän tarvitse tuntea tietokannan rakennetta tarkalleen, ja tietokannan sisällön ja rakenteen tutkiminen helpottuu.

Tässä tutkielmassa tutkittiin käyttäjäkokeen avulla, miten käyttäjät onnistuvat PathSQL-kielen käytössä verrattuna SQL-kieleen kyselyissä, jotka kohdistuvat hierarkkiseen dataan. Koetehtävät olivat molemmilla kielillä samat ja kohdistuivat samaan dataan. Koeasetelma ei ollut kuitenkaan molempien kielten osalta tasa-arvoinen, sillä koehenkilöinä käytettiin tietojenkäsittelytieteen kurssin opiskelijoita, joilla oli aiempaa kokemusta SQL-kielestä vähintään yhden kurssin verran (useimmilla jopa enemmän), kun taas PathSQL-kieli esiteltiin opiskelijoille lyhyesti kokeen yhteydessä (sekä aiemmalla luennolla, jolle osallistuminen ja luentomateriaaliin perehtyminen oli vapaaehtoista).

Kokeen tulosten perusteella vaikuttaa siltä, että käyttäjät suoriutuvat hierarkkiseen dataan kohdistuvista kyselyistä PathSQL-kielellä nopeammin kuin SQL-kielellä. Ero on erityisen selvä silloin, kun PathSQL-kielen kysely on osattu tehdä oikein. Sen sijaan koehenkilöt ratkaisivat tehtävät oikein vain hieman useammin PathSQL-kielellä kuin SQL-kielellä, eikä ero ole tilastollisesti merkitsevä. Koehenkilöiden aiempaa menestystä tietokantoihin liittyvillä kursseilla ja kokeen tulosta verrattaessa vaikuttaa siltä, että aiempi hyvä kurssimenestys tietokantakursseilla on yhteydessä PathSQL-kielen nopeaan oppimiseen. Ei ole kuitenkaan täysin varmaa, mistä tämä johtuu: syynä voi olla esimerkiksi opiskelumotivaatio, lahjakkuus, tai SQL-kielen hyvä osaaminen.

Kokeen tehtävistä PathSQL-tehtävät 8 ja 9 sekä SQL-tehtävät 16 ja 17 olivat koehenkilöille erityisen vaikeita. Tehtävät 8 ja 16 sekä tehtävät 9 ja 17 vastasivat tehtävänannoltaan toisiaan. Puolet koehenkilöistä epäonnistui näissä kaikissa tehtävissä. Lisäksi näistä jokaisessa tehtävässä enemmistö koehenkilöiden vastauksista oli väärin. Näille kaikille tehtäville oli ominaista se, että niissä hierarkian alkurelaatiosta loppurelaatioon johti useampi kuin yksi polku: PathSQL-kyselyissä

tämä tarkoitti //-operaattorin oikeanlaista käyttöä polkuilmaisuissa ja SQL-kyselyissä pitkää ja moniosaista kyselyä.

Koehenkilöiden virheellisten PathSQL-kyselyiden virheet luokiteltiin syntaksivirheisiin, jotka ovat kielen syntaksin vastaisia, ja semanttisiin virheisiin, jotka ovat muodoltaan oikeaa PathSQL-kieltä mutta jotka tuottavat väärän tuloksen. Syntaksivirheitä oli neljää eri tyyppiä, ja semanttisia virheitä seitsemää eri tyyppiä. Osalla virhetyypeistä oli lisäksi alatyyppejä. Semanttiset virheet olivat yleisempiä kuin syntaksivirheet: yli puolet kaikista koehenkilöiden PathSQL-tehtävien ratkaisuyrityksistä sisälsi vähintään yhden semanttisen virheen. Yleisin virhe oli polun haaroittamiseen tarkoitettujen sulkumerkkien virheellinen käyttö: tämän virheen teki vähintään kerran enemmistö koehenkilöistä. Erityisen yleisiä nämä virheet olivat koehenkilöille haastavimmissa tehtävissä 8 ja 9, joissa alkurelaatiosta loppurelaatioon johti useampi kuin yksi polku, ja joissa polkujen haaroittamisen sijaan olisi pitänyt käyttää //-operaattoria.

Mahdollisissa tulevissa käyttäjäkokeissa tulisi kiinnittää huomiota joihinkin koejärjestelyihin, jotka saattavat vaikuttaa tulokseen: esimerkiksi järjestelmän antama palaute, tietokannan tauluissa käytetty kieli ja tehtävänantojen muotoilu voivat olla tällaisia asioita. Joko täysin ohjelmointitaidottomien ja kyselykieliä osaamattomien noviisikäyttäjien tai polkuorientoituneita kieliä (kuten XPath) aiemmin käyttäneiden koehenkilöiden käyttö saattaisi tuottaa kiinnostavia vertailukohteita tämän tutkielman käyttäjäkokeelle. Kielen jatkokehityksessä ja mahdollisessa opetuksessa kannattaa ottaa huomioon käyttäjäryhmä, jolle kieli on suunnattu: jos kielen käyttäjillä on aiempaa kokemusta esimerkiksi SQL-kielen käytöstä, se saattaa joiltain osin hankaloittaa mutta myös monella tapaa helpottaa PathSQL-kielen oppimista.

Viiteluettelo

[Brass and Goldberg, 2005] Stefan Brass and Christian Goldberg, Semantic errors in SQL queries: A quite complete list. *The Journal of Systems and Software* **79** (2006), 630-644.

[Buneman et al., 1996] Peter Buneman, Susan Davidson, Gerd Hillebrand and Dan Suciu, A query language and optimization techniques for unstructured data. In: *SIGMOD '96 Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, 505-516.

[Carey et al., 1988] Michael J. Carey, David J. DeWitt and Scott L. Vandenberg, A data model and query language for EXODUS. In: *SIGMOD '88 Proceedings of the 1988 ACM SIGMOD international conference on Management of data*, 413-423.

[Catarci and Santucci, 1995] Tiziana Catarci and Giuseppe Santucci, Diagrammatic vs textual query languages: a comparative experiment. In: *Visual Database Systems* 3, Springer US, 1995, 69-83.

[Cattell & Barry, 2000] R.G.G. Cattell, Douglas K. Barry, *The Object Data Standard: ODMG 3.0*. Morgan Kaufmann, 2000.

[Chamberlin and Boyce, 1974] Donald D. Chamberlin, Raymond F. Boyce, SEQUEL: A Structured English Query Language. In: *Proceedings of the 1974 ACM SIGFIDET Workshop on Data Description, Access and Control (Association for Computing Machinery),* 249–64.

[Cluet, 1998] Sophie Cluet, Designing OQL: Allowing objects to be queried. Information systems **23**, 5 (1998), 279-305.

[Codd, 1970] Edgar F. Codd, A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM* **13**,6 (1970), 377–387.

[David, 2003] Michael M. David, ANSI SQL hierarchical processing can fully integrate native XML. *ACM SIGMOD Record* **32**,1 (2003), 41-46.

[Deutsch et al., 1999] Alin Deutsch, Mary Fernandez, Daniela Florescu, Alon Levy and Dan Suciu, A query language for XML. Computer networks **31,** 11 (1999), 1155-1169.

[Elmasri & Navathe, 1989] Ramez Elmasri and Shamkant B. Navathe, *Fundamentals of Database Systems*. Benjamin Cummings, 1989.

[Fuhr & Großjohann, 2001] Norbert Fuhr and Kai Großjohann, XIRQL: A query language for information retrieval in XML documents. In: *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 172-180.

[Goldman et al., 1999] Roy Goldman, Jason McHugh and Jennifer Widom, From semistructured data to XML: Migrating the Lore data model and query language. Available as *http://ilpubs.stanford.edu:8090/409/1/1999-53.pdf*.

[Graaumans, 2005] Joris Petrus Maria Graaumans, *Usability of XML Query Languages*. Instituut voor Informatica en Informatiekunde, Universiteit Utrecht, 2005.

[Han et al., 2007] Jiawei Han, Hong Cheng, Dong Xin and Xifeng Yan, Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery* **15**, 1 (2007), 55-86.

[Hua & Tripathy, 1994] Kien A. Hua and Chinmoy Tripathy, Object Skeletons: An efficient navigation structure for object-oriented database systems. In: *Data Engineering, 1994. Proceedings. 10th International Conference*, 508 – 517.

[Jaakkola and Thalheim, 2003] Hannu Jaakkola and Bernhard Thalheim, Visual SQL – High-Quality ER-Based Query Treatment. In: *Conceptual Modeling for Novel Application Domains*, Springer Berlin Heidelberg, 2003, 129-139.

[Junkkari, 2005] Marko Junkkari, PSE: An object-oriented representation for modeling and managing part-of relationships. *Journal of Intelligent Information Systems* **25**, 2 (2005), 131-157.

[ISO/IEC 14977, 1996] International standard, *Information technology - Syntactic metalanguage - Extended BNF*, ISO/IEC, 1996.

[Motschnig-Pitrik & Kaasbøll, 1999] Renate Motschnig-Pitrik and Jens Kaasbøll, Part-whole relationship categories and their application in object-oriented analysis. IEEE Transactions on Knowledge and Data Engineering **11**, 5 (1999), 779-797.

[Niemi & Järvelin, 1995] Timo Niemi & Kalervo Järvelin, A straightforward NF^2 relational interface with applications in information retrieval. *Information processing & management* **31**, 2 (1995), 215-231.

[Niemi et al., 2004] Timo Niemi, Marko Junkkari, Kalervo Järvelin, and Samu Viita, Advanced query language for manipulating complex entities. *Information Processing and Management* **40** (2004), 869–889.

[Reisner, 1981] Phyllis Reisner, Human factors studies of database query languages: a survey and assessment. *ACM Computing Surveys* **13**, 1 (March 1981), 13-31.

[Reisner et al., 1975] Phyllis Reisner, Raymond F. Boyce, and Donald D. Chamberlin, Human factors evaluation of two data base query languages – Square and Sequel. In: *Proc. of National Computer Conference*, 447-452.

[Roth et al., 1988] Mark A. Roth, Herry F. Korth, and Abraham Silberschatz, Extended algebra and calculus for nested relational databases. *ACM Transactions on Database Systems* **13**,4 (1988), 389-417.

[Sengupta & Dillon, 2006] Arijit Sengupta and Andrew Dillon, Query by templates: Using the shape of information to search next-generation databases. *IEEE Transactions on Professional Communication*, **49,** 2 (2006), 128-144.

[Thomas and Gould, 1975] John C.Thomas and John D. Gould, A psychological study of query by example. In: *Proc. of National Computer Conference*, 439-445.

[Vainio and Junkkari, 2014] Johanna Vainio and Marko Junkkari, SQL based semantics for path expressions over hierarchical data in relational databases. *Journal of Information Science* **40**,3 (June 2014), 293-312.

[Webb, 2010] Geoffrey I. Webb, Self-sufficient itemsets: An approach to screening potentially interesting associations between items. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **4,** 1 (2010), 3:1-3:20.

[Webb, 2012] Geoffrey I. Webb, *Magnum Opus 4.6.3 Tutorial Introduction*. G. I. Webb & Associates Pty Ltd., 2012.

[Weiand, 2010] Klara Weiand, *Keyword-based Querying for the Social Semantic Web.* Fakultät für Mathematik, Informatik und Statistik, Ludwig-Maximilians-Universität München, 2010.

[Welty and Stemple, 1981] Charles Welty and David W. Stemple, Human factors comparison of a procedural and a nonprocedural query language. *ACM Transactions on Database Systems* **6**, 4 (Dec. 1981), 626-649.

[Wilcoxon, 1945] Frank Wilcoxon, Individual comparisons by ranking methods, *Biometrics Bulletin*, **1**, 6. (Dec. 1945), 80-83.

[Winston et al., 1987] Morton E. Winston, Roger Chaffin and Douglas Herrmann, A taxonomy of part-whole relations. Cognitive science **11**, 4 (1987), 417-444.

[World Wide Web Consortium, 2005] The XML data model, http://www.w3.org/XML/Datamodel.html. Katsottu 7.10.2014.

[World Wide Web Consortium, 2008] XML 1.0 Specification, http://www.w3.org/TR/REC-xml/. Katsottu 15.9.2014.

[World Wide Web Consortium, 2014 a] XPath 3.0 W3C Recommendation, http://www.w3.org/TR/xpath-30/. Katsottu 15.9.2014.

[World Wide Web Consortium, 2014 b] XQuery 3.0 Recommendation, http://www.w3.org/TR/xquery-30/. Katsottu 15.9.2014.

[Yen and Scamell, 1993] Minnie Yi-Miin Yen and Richard W. Scamell, A human factors experimental comparison of SQL and QBE. *IEEE Transactions on Software Engineering* **19**, 4 (1993), 390-409.

Koehenkilöille koetilaisuudessa annettu esimerkkimateriaali

Polkuorientoitunut SQL (PathSQL)

- Kyselyt hierarkkiseen dataan
- From-osassa annetaan polkuilmauksilla rakenne, johon kysely kohdistuu

Esimerkki

- Polkupyörä sisältää välittömät komponentit: *runko*, *ohjaus* ja *rengas* (takarengas)
- Ohjaus koostuu *akselista*, *ohjaustangosta* ja *renkaasta* (eturengas)

Kysely 1: Valitse polkupyörien merkit ja niihin liittyvät runkojen numerot.

```
SELECT polkupyora.merkki, runko.numero
   FROM polkupyora/runko
```

- FROM-osaan valitaan siis yksiaskelinen polku (myös `//` toimii)

Kysely 2: Valitse merkkiä 'jopo' olevien polkupyörien runkojen numerot.

```
SELECT runko.numero
   FROM polkupyora/runko
   WHERE polkupyora.merkki = 'jopo'
```

- Samoin kuin edellä, mutta nyt polkupyörän attribuutille annetaan valintaehto
- Valintaehto voidaan siis antaa mille tahansa polun jäsenelle

Kysely 3. Valitse merkkiä 'helkama' olevien polkupyörien renkaiden halkaisijat.

```
SELECT rengas.halkaisija
   FROM polkupyora//rengas
   WHERE polkupyora.merkki = 'helkama'
```

- Moni-askelisella polulla `//` valitaan kaikki polut polkupyörästä renkaaseen

- Esimerkissä tämä tarkoittaa polkuja `polkupyörä/rengas` sekä `polkupyörä/ohjaus/rengas`

- Kysely kohdistuu siis molempiin polkuihin

Kysely 4: Mitä ohjaustangon mallia käytetään missäkin rungossa?

```
SELECT tanko.malli, runko.numero
   FROM polkupyora(/ohjaus/tanko,/runko)
```

- Vastaukseen tarvitaan tietoa kahdesta haarasta: `polkupyora/ohjaus/tanko` ja `polkupyora/runko`
- Polun haaroittuminen ilmaistaan suluilla

- Myös ilmaisu `polkupyora(//tanko,/runko)` on mahdollinen

Esimerkkejä koedatan käsittelyssä käytetyistä SQL-kyselyistä

Monennellako ratkaisuyrityskerralla vähintään, enintään ja keskimäärin saatiin oikea vastaus ja tehtävien suoritukseen vähintään, enintään ja keskimäärin käytetty aika:

```
SELECT onnistuiko, ajat.tehtava,
       MIN(ajat.aika) AS minimiaika,
         MAX(ajat.aika) AS maksimiaika,
         AVG(ajat.aika) AS keskiaika,
       MIN(ajat.yritystenlkm) AS minimilkm,
         MAX(ajat.yritystenlkm) AS maksimilkm,
         AVG(ajat.yritystenlkm) AS keskilkm

FROM

(

SELECT tiop.kayttaja.id, tiop.tehtava.id as tehtava,
SUM(EXTRACT (SECOND FROM tiop.ratkaisuyritys.loppui-
tiop.ratkaisuyritys.alkoi)+
EXTRACT (MINUTE FROM tiop.ratkaisuyritys.loppui-
tiop.ratkaisuyritys.alkoi)*60) AS aika,
COUNT(*) AS yritystenlkm,
MIN(tiop.ratkaisuyritys.onnistui) AS orig_onnistui,
CASE WHEN MIN(tiop.ratkaisuyritys.onnistui) = 0 THEN 1
     ELSE 0
END    AS onnistuiko

FROM tiop.kayttaja, tiop.sessio, tiop.tehtavasarja,
tiop.tehtava_tehtavasarja, tiop.tehtava, tiop.ratkaisuyritys

WHERE tiop.kayttaja.id=tiop.sessio.kayttaja
AND tiop.sessio.tehtavasarja=tiop.tehtavasarja.id AND
tiop.tehtavasarja.id=tiop.tehtava_tehtavasarja.tehtavasarja
AND tiop.tehtava_tehtavasarja.tehtava=tiop.tehtava.id AND
tiop.tehtava.id=tiop.ratkaisuyritys.tehtava AND
tiop.ratkaisuyritys.sessio=tiop.sessio.id

GROUP BY tiop.kayttaja.id, tiop.tehtava.id

) AS ajat

GROUP BY ajat.tehtava, ajat.onnistuiko

ORDER BY ajat.onnistuiko desc, ajat.tehtava;
```

Käyttäjien tehtävien suorittamiseen käyttämät ajat:

```
SELECT tiop.kayttaja.id, tiop.tehtava.id as tehtava,
SUM(EXTRACT (SECOND FROM tiop.ratkaisuyritys.loppui-
tiop.ratkaisuyritys.alkoi)+
EXTRACT (MINUTE FROM tiop.ratkaisuyritys.loppui-
tiop.ratkaisuyritys.alkoi)*60) AS aika,
COUNT(*) AS yritystenlkm,
MIN(tiop.ratkaisuyritys.onnistui) AS orig_onnistui,
CASE WHEN MIN(tiop.ratkaisuyritys.onnistui) = 0 THEN 1
     ELSE 0
```

```
END    AS onnistuiko

FROM tiop.kayttaja, tiop.sessio, tiop.tehtavasarja,
tiop.tehtava_tehtavasarja, tiop.tehtava, tiop.ratkaisuyritys

WHERE tiop.kayttaja.id=tiop.sessio.kayttaja
AND tiop.sessio.tehtavasarja=tiop.tehtavasarja.id
AND tiop.tehtavasarja.id=tiop.tehtava_tehtavasarja.tehtavasarja
AND tiop.tehtava_tehtavasarja.tehtava=tiop.tehtava.id
AND tiop.tehtava.id=tiop.ratkaisuyritys.tehtava
AND tiop.ratkaisuyritys.sessio=tiop.sessio.id

GROUP BY tiop.kayttaja.id, tiop.tehtava.id

ORDER BY tiop.kayttaja.id, tiop.tehtava.id;
```

Onnistuneiden tehtävien lukumäärät:

```
SELECT tiop.tehtava.id as tehtava, COUNT(tiop.ratkaisuyritys.id) as
onnistuneet

FROM tiop.kayttaja, tiop.sessio, tiop.tehtavasarja,
tiop.tehtava_tehtavasarja, tiop.tehtava, tiop.ratkaisuyritys

WHERE tiop.kayttaja.id=tiop.sessio.kayttaja
AND tiop.sessio.tehtavasarja=tiop.tehtavasarja.id
AND tiop.tehtavasarja.id=tiop.tehtava_tehtavasarja.tehtavasarja
AND tiop.tehtava_tehtavasarja.tehtava=tiop.tehtava.id
AND tiop.ratkaisuyritys.sessio=tiop.sessio.id
AND tiop.kayttaja.id>9
AND tiop.tehtava.id<20
AND tiop.ratkaisuyritys.onnistui=0

GROUP BY tiop.tehtava.id

ORDER BY tiop.tehtava.id;
```

Virheelliset PathSQL-ratkaisuyritykset:

```
SELECT tiop.tehtava.id as tehtava, tiop.kayttaja.id as kayttaja,
tiop.tehtava.tehtavananto, tiop.tehtava.ratkaisu,
tiop.ratkaisuyritys.syote, tiop.ratkaisuyritys.nro

FROM tiop.kayttaja, tiop.sessio, tiop.tehtavasarja,
tiop.tehtava_tehtavasarja, tiop.tehtava, tiop.ratkaisuyritys

WHERE tiop.kayttaja.id=tiop.sessio.kayttaja
AND tiop.sessio.tehtavasarja=tiop.tehtavasarja.id
AND tiop.tehtavasarja.id=tiop.tehtava_tehtavasarja.tehtavasarja
AND tiop.tehtava_tehtavasarja.tehtava=tiop.tehtava.id
AND tiop.tehtava.id=tiop.ratkaisuyritys.tehtava
AND tiop.ratkaisuyritys.sessio=tiop.sessio.id
AND NOT tiop.ratkaisuyritys.onnistui=0
AND tiop.tehtava.tyyppi='pathSQL'

ORDER BY tiop.tehtava.id, tiop.kayttaja.id;
```

Niiden käyttäjien lukumäärä tehtäväpareittain, jotka onnistuivat ratkaisemaan oikein PathSQL-tehtävän mutta eivät SQL-tehtävää:

```
SELECT t1.tehtava, COUNT(t1.kayttaja)

FROM

(
SELECT tiop.tehtava.id as tehtava, tiop.kayttaja.id as kayttaja

FROM tiop.kayttaja, tiop.sessio, tiop.tehtavasarja,
tiop.tehtava_tehtavasarja, tiop.tehtava, tiop.ratkaisuyritys

WHERE tiop.kayttaja.id=tiop.sessio.kayttaja AND
tiop.sessio.tehtavasarja=tiop.tehtavasarja.id AND
tiop.tehtavasarja.id=tiop.tehtava_tehtavasarja.tehtavasarja AND
tiop.tehtava_tehtavasarja.tehtava=tiop.tehtava.id AND
tiop.tehtava.id=tiop.ratkaisuyritys.tehtava AND
tiop.ratkaisuyritys.sessio=tiop.sessio.id AND
tiop.ratkaisuyritys.onnistui=0

) t1

JOIN

(
SELECT tiop.tehtava.id as tehtava, tiop.kayttaja.id as kayttaja,
CASE WHEN MIN(tiop.ratkaisuyritys.onnistui) = 0 THEN 1
     ELSE 0
END    AS onnistuiko

FROM tiop.kayttaja, tiop.sessio, tiop.tehtavasarja,
tiop.tehtava_tehtavasarja, tiop.tehtava, tiop.ratkaisuyritys

WHERE tiop.kayttaja.id=tiop.sessio.kayttaja AND
tiop.sessio.tehtavasarja=tiop.tehtavasarja.id AND
tiop.tehtavasarja.id=tiop.tehtava_tehtavasarja.tehtavasarja AND
tiop.tehtava_tehtavasarja.tehtava=tiop.tehtava.id AND
tiop.tehtava.id=tiop.ratkaisuyritys.tehtava AND
tiop.ratkaisuyritys.sessio=tiop.sessio.id

GROUP BY tiop.kayttaja.id, tiop.tehtava.id

) t2

ON (t1.tehtava = 4 AND t2.tehtava = 12 OR
    t1.tehtava = 5 AND t2.tehtava = 13 OR
      t1.tehtava = 6 AND t2.tehtava = 14 OR
      t1.tehtava = 7 AND t2.tehtava = 15 OR
      t1.tehtava = 8 AND t2.tehtava = 16 OR
      t1.tehtava = 9 AND t2.tehtava = 17 OR
      t1.tehtava = 10 AND t2.tehtava = 18) AND
      t1.kayttaja = t2.kayttaja AND
      t2.onnistuiko=0

GROUP BY t1.tehtava

ORDER BY t1.tehtava;
```